Unclassified
Security Classification

AD 735 145

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) University of Georgia	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP Unclassified

3. REPORT TITLE

INTERACTIVE QUANTAL ANALYSIS

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

5. AUTHOR(S) *(First name, middle initial, last name)*

JUDITH H. ISHEE

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
September, 1971	127	10

8a. CONTRACT OR GRANT NO. N00014-69-A-~~042-261~~ 0423 b. PROJECT NO. NR 042-261 c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) Technical Report No. 74 Dept. of Statistics and Computer Science 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)* No. 16 Themis

10. DISTRIBUTION STATEMENT

Reproduction in whole or in part is permitted for any purpose of the United States Government.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Office of Naval Research Washington, D. C.

13. ABSTRACT

This report describes development and implementation of a conversational computer program for the statistical analysis of biological assays and related quantal data. The program is designed for use by scientists who need not possess more than an elementary knowledge of statistics or computation. A graphics console (CRT display, typewriter, function keyboard) serves as the medium of conversation. Questions are displayed to the user. On the basis of data supplied by him, the appropriate analysis is performed and results are displayed graphically and numerically.

DD FORM 1 NOV 65 1473 (PAGE 1)
S/N 0101-807-6811

Unclassified
Security Classification

A-31408

KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
STATISTICS						
REGRESSION						
BIO-ASSAY						
QUANTAL ANALYSIS						
INTERACTIVE COMPUTATION						
COMPUTER GRAPHICS						
NONLINEAR ESTIMATION						
PROBIT ANALYSIS						
GROWTH CURVES						
LEARNING MODELS						

THEMIS REPORT NUMBER 16

TECHNICAL REPORT NUMBER 74

INTERACTIVE QUANTAL ANALYSIS

JUDITH H. ISHEE

Reproduction in whole or in part is permitted for any purpose of the United States Government. This Research was supported, in part, by the Office of Naval Research, Contract Number N00014-69-A-0423, NR 042-261.

ROLF BARGMANN
PRINCIPAL INVESTIGATOR

THE UNIVERSITY OF GEORGIA
DEPARTMENT OF STATISTICS AND COMPUTER SCIENCE
ATHENS, GEORGIA 30601

SEPTEMBER 1971

ABSTRACT

Several programs are in existence which perform quantal analysis, but such programs were designed for batch processing. The following report is a package of computer programs for the IBM 2250, with documentation, which enables the layman to interact with the system. The program, entitled "QUANTAL" is written in FORTRAN IV with extensive use of the GRAF (Graphics Addition to FORTRAN) routines. QUANTAL is written in subroutine form within an overlay structure which retains the entire program size to within 64K bytes. Communication between subroutines is attained by extensive use of COMMON variables and direct disk storage. The method of analysis follows the usual techniques of maximum-likelihood iteration, with an approximate Newton technique.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I

INTRODUCTION

A great variety of experiments, especially those in biological assay, establish a relationship between dosage (a concomitant variable) and a quantal response. The latter consists of records of successes or failures under a given dosage. The quantal response is often "death," but may be any other easily recognized change in an experimental subject. For example, insecticides may be assayed by assigning batches to doses of a test preparation -- here the quantal response is the number of insects killed in each batch. In the physical or social sciences, the concomitant variable is usually "time." At specific intervals a record is kept about the number of subjects that completed a task, or the number of experimental runs that have proceeded to completion in a reaction. These are the quantal responses.

This thesis first summarizes a few of the well known basic results for the analysis of quantal response data. Essentially, the approach presented by Finney [3], [4] is followed, including the notation, somewhat peculiar for statisticians, which uses capital letters for parameters. Chapter II also includes a discussion of the various algorithms as they apply to a few preferred growth functions (Probit, Logistic, Arc Sine, Log-Log, Weibull -- see [1], [3]).

The implementation, including user's guide, description, and illustration, is contained in Chapter III. This is an example of an interactive program designed for the user whose expert knowledge in Statistics

or Computations may be limited. In fact, the present program has been used in the research work of entomologists at the University of Georgia. This part of the work is directed toward the implementation of the University of Georgia's development of Statistical Programs for Lay Users, under the THEMIS program.

Detailed documentation of the various computer programs, the communication with the Graphics Monitor System, and the overlay structure, are presented in the last chapter.

A statistician desiring to use the computer for the analysis of experimental data has a variety of programs, procedures, and languages at his disposal. The majority of these programs (BMD [2], GENSTAT [9], OMNITAB [6], and SSP [8], to name a few) require some degree of technical understanding. Often a user must even supply a complete function program or a subroutine. Thus, unless the statistician possesses enough experience in programming and data processing to prepare his input data, he must seek the advice of one with such experience.

The interactive console programs, on the other hand, are designed so that by merely answering questions and filling in blanks a layman, who possesses little or no knowledge in the fields of Statistics or Computation, may perform the analysis on his particular data. The QUANTAL program also affords the layman the opportunity to change his data input (by adding or deleting batches, or performing some transformation on the x-dosage) and immediately see the results. This eliminates the necessity of re-submission of the job deck under the batch mode and thus increases the overall effectiveness of turnaround.

CHAPTER II

COMPUTATIONAL ALGORITHMS

In a biological assay, a stimulus (for example a vitamin, a drug, a physical force) is applied to a subject (for example an animal, a plant, or a single cell) at an intensity specified in units of concentration, weight, time or other appropriate measure and under environmental conditions as carefully controlled as is practicable, as a result of which a response is produced by the subject.

The derivation of iterative maximum-likelihood estimates follows familiar lines (see, e.g., Finney [3]). Let us assume that there are k classes of subjects and n_i is the size of class i. Suppose r_i denotes the number of "successful" individuals in class i and P_i denotes the probability that a subject in class i has made the transition. The probability of exactly r_i responding is, the Binomial Distribution of probabilities

$$n_i!/r_i!(n_i - r_i)! \qquad P_i^{r_i} Q_i^{n_i - r_i}$$

where $Q_i = 1 - P_i$. Then the logarithm of the likelihood function of $(r_1, r_2, \ldots r_k)$ is

$$\log L = \sum_{i=1}^{k} [\log \binom{n_i}{r_i} + r_i \log P_i + (n_i - r_i)\log(1 - P_i)] \qquad (2.1)$$

If we let p_i denote the observed proportion of successes in class i ($p_i = r_i/n_i$), we find that

$$\partial \log L/\partial P_i = n_i(p_i - P_i)/P_i(1 - P_i) \tag{2.2}$$

As is customary in bio-assay analysis, we use capital letters to denote expected values and small letters to denote observed values.

The dosage-response analysis proceeds as follows. A monotonic (growth) function is defined

$$P_i = \Phi(Y_i) \tag{2.3}$$

which is frequently a cumulative distribution function. It is then assumed that Y_i follows a linear regression model

$$Y_i = \alpha + \beta x_i, \tag{2.4a}$$

where x_i is the dosage (or some convenient transform).

$$Y_i = \Phi^{-1}(P_i)$$

is a "percentage point" under this c.d.f. It is a linear function of x_i.

Let $$Z_i = dP_i/dY_i \tag{2.5}$$

Let $\bar{x}$ denote a conveniently chosen constant (usually a weighted mean of the x_i) and let

$$\mu = \alpha + \beta\bar{x}, \quad \text{so that} \quad Y_i = \mu + \beta(x_i - \bar{x}). \tag{2.4b}$$

Then

$$\partial \log L/\partial\mu = \sum_{i=1}^{k} [\partial \log L/\partial P_i][dP_i/dY_i][\partial Y_i/\partial\mu]$$

$$= \sum_{i=1}^{k} [n_i(p_i - P_i)Z_i]/[P_i(1 - P_i)] \tag{2.6a}$$

and, similarly,

$$\partial \log L/\partial\beta = \sum_{i=1}^{k} [n_i(p_i - P_i)Z_i(x_i - \overline{x})]/[P_i(1 - P_i)] \tag{2.6b}$$

Expressions (2.6a) and (2.6b) must be equated to zero and solved for μ and β (involved in each Y_i, hence in each P_i and Z_i). This is customarily done by an approximate Newton iteration where $y_i = \Phi^{-1}(p_i)$ is used as the first approximation to Y_i.

If we define

$$W_i = Z_i^2/[P_i(1 - P_i)], \tag{2.7}$$

(2.6a) and (2.6b) can be rewritten

$$\partial \log L/\partial\mu = \sum_{i=1}^{k} n_i W_i[(p_i - P_i)/Z_i] \tag{2.8a}$$

and

$$\partial \log L/\partial\beta = \sum_{i=1}^{k} n_i W_i[(p_i - P_i)/Z_i](x_i - \overline{x}) \tag{2.8b}$$

Now,

$$\partial^2 \log L/\partial\mu^2 = \sum_{i=1}^{k} \{[\partial(\partial \log L/\partial\mu)/\partial P_i][\partial P_i/\partial Y_i][\partial Y_i/\partial\mu]$$

$$+ [\partial(\partial \log L/\partial\mu)/\partial Z_i][\partial Z_i/\partial Y_i][\partial Y_i/\partial\mu]\}.$$

Now

$$\partial(\log L/\partial\mu)/\partial P_i = -(n_i Z_i)/[P_i(1 - P_i)]$$

$$-[n_i(p_i - P_i)Z_i(1 - 2P_i)]/[P_i(1 - P_i)]^2$$

The essential trick in the approximate Newton iteration (see Fisher [5]) consists of ignoring small terms in the second derivative. Thereby the iterative procedure is, admittedly, slower than the complete Newton procedure. However, since no approximation is employed in the first derivatives, the final result will still be exact when the process has converged. In the above example, $(p_i - P_i)$ is a relatively small term and will be disregarded in the second derivative. Also,

$$\partial(\partial\log L/\partial\mu)/\partial Z_i$$

involves such a difference $(p_i - P_i)$ and hence will be disregarded. With this simplification one obtains the approximate value

$$\partial^2\log L/\partial\mu^2 \simeq -\sum_{i=1}^{k} n_i\{Z_i^2/[P_i(1 - P_i)]\} = -\sum_{i=1}^{k} n_i W_i. \tag{2.9a}$$

Similarly,

$$\partial^2\log L/\partial\beta^2 \simeq -\sum_{i=1}^{k} n_i W_i(x_i - \bar{x})^2 \tag{2.9b}$$

and

$$\partial^2\log L/\partial\mu\partial\beta \simeq -\sum_{i=1}^{k} n_i W_i(x_i - \bar{x}) \tag{2.9c}$$

$$= 0 \text{ if } \bar{x} \text{ is chosen as}$$

$$\sum_{i=1}^{k} n_i W_i x_i / \sum_{i=1}^{k} n_i W_i .$$

With this simplification, the Hessian matrix of second derivatives is diagonal. Hence corrections can be made on each of the two unknowns separately, by substitution into the customary Newton formula.

$$\mu_1 = \mu_0 + \sum_{i=1}^{k} n_i W_i [(p_i - P_i)/Z_i] / \sum_{i=1}^{k} n_i W_i$$

and

$$\beta_1 = \beta_0 + \sum_{i=1}^{k} n_i W_i [(p_i - P_i)/Z_i](x_i - \bar{x}) / \sum_{i=1}^{k} n_i W_i (x_i - \bar{x})^2 .$$

If we estimated μ_0 and β_0 by weighted regression on Y_i given x_i, with the weights nW, we could combine the μ_0 and the sum into a single sum

$$\mu_1 = \sum_{i=1}^{k} n_i W_i [Y_i + (p_i - P_i)/Z_i](x_i - \bar{x}) / \sum_{i=1}^{k} n_i W_i (x_i - \bar{x})^2 ,$$

and, analogously

$$\beta_1 = \sum_{i=1}^{k} n_i W_i [Y_i + (p_i - P_i)/Z_i](x_i - \bar{x}) / \sum_{i=1}^{k} n_i W_i (x_i - \bar{x})^2$$

But these are estimates of μ_1 and β_1 of a weighted regression analysis, between the x_i and the quantities $[Y_i + (p_i - P_i)/Z_i]$. The latter are called "working" quantities. Thus, the approximate Newton iteration employs the same technique as the iterated (weighted regression) method, with y_i replaced by the working quantities, $y_i + (p_i - P_i)/Z_i$.

The c.d.f. most frequently employed in the analysis of such data is the Normal distribution function, often with the mean placed at 5 to avoid negative numbers. This is called the Probit transformation. In detail, the following steps are required:

Data for the analysis are, for each of k batches, x_i, the stimulus or dosage applied; n_i, the number of subjects tested at each dose; and r_i, the number of subjects who responded at each dose.

The first cycle proceeds as follows:

(a) Compute the proportion of subjects responding to various levels of a stimulus $p_i = r_i/n_i$ where $i = 1,2,\ldots k$ levels of a stimulus.

(b) Obtain y_i, the empirical probit for each dosage level; $y_i = \Phi^{-1}(p_i)$, where $\Phi^{-1}(p_i)$ is the inverse normal distribution (e.g., $\Phi^{-1}(.975) = 1.96$ or 6.96 if 5 is added).

(c) Initialize the working probit by setting $y_i^* = y_i$ and the weighting coefficient for the probit analysis, by using $W_i = n_i$.

(d) From the weighted regression scheme, obtain a and b, estimates of α and β in the equation $y_i = \alpha + \beta x_i$.

$$b = S_{xy}^{(w)}/S_{xx}^{(w)} \quad \text{where}$$

$$S_{xy}^{(w)} = \sum_{i=1}^{k} w_i x_i y_i^* - [(\sum_{i=1}^{k} w_i x_i)(\sum_{i=1}^{k} w_i y_i^*)]/\sum_{i=1}^{k} w_i$$

and

$$S_{xx}^{(w)} = \sum_{i=1}^{k} w_i x_i^2 - (\sum_{i=1}^{k} w_i x_i)^2/\sum_{i=1}^{k} w_i.$$

$a = \bar{y} - b\bar{x}$ where

$$\bar{y} = \sum_{i=1}^{k} w_i y_i / \sum_{i=1}^{k} w_i \quad \text{and}$$

$$\bar{x} = \sum_{i=1}^{k} w_i x_i / \sum_{i=1}^{k} w_i .$$

(Note that w_i is $n_i W_i$ here).

(e) Compute values of expected probit using the initial estimates of a and b: $Y_i = a + b\, x_i$ where $i = 1,2,\ldots k$.

(f) Calculate P_i, the predicted proportion of success at each dose. Here, $P_i = \Phi(Y_i)$[or, possibly, $Y_i - 5$], is the cumulative normal distribution value corresponding to Y_i.

(g) Obtain $Z_i = \phi(Y_i)$, the ordinate under the normal distribution curve, for each Y_i.

(h) Corresponding to each value of Y, compute the following:

$W_i = Z_i^2/[P_i(1 - P_i)]$ and $y_i^* = Y_i + (p_i - P_i)/Z_i$.

(i) Return to step (d) and begin a new iteration with $a = \alpha$ and $b = \beta$ until the relative difference between two consecutive values of α or β are small. Relative error of 10^{-4}, or 5 iterations, are employed in this program.

Other models may be substituted in this analysis. The following equations for P, y, and Z should be substituted in steps (b), (f), and (g):

The logistic function, $dn/dt = \lambda n(c - n)$, where c is the ceiling and n is some population size, is widely used to represent such phenomena as population growth. $P = e^Y/(1 + e^Y) = 1/(1 + e^{-Y})$ gives the metametric equation and $Y = \log[P/(1 - P)]$. So that $Z = dP/dY = e^Y/(1 + e^Y)^2$.

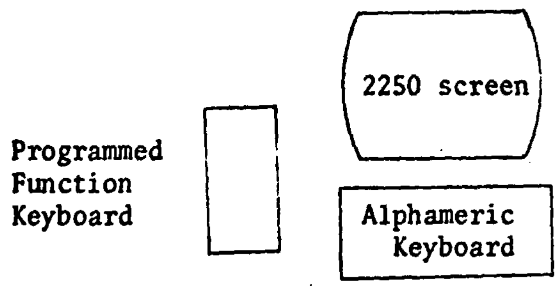
The log-log or "Gumpertz" function is used when reproduction inhibits itself. The metametric equation is $P = e^{-\epsilon^{-Y}}$ and $Y = -\log(-\log P)$. Also $Z = dP/dY = -P \log P$.

For the Arc Sine function, $P = \frac{1}{2}(1 + \sin Y)$, $Y = \text{Arc sin } (2P - 1)$, and $Z = \frac{1}{2}\cos Y$.

Finally, for the Weibull function, $P = 1 - e^{-Y^r}$, $Y = [-\log (1 - P)]^{1/r}$, $Z = r\, Y^{r-1} e^{-Y^r}$.

CHAPTER III

THE INTERACTIVE PROGRAM



The diagram above displays the position of the IBM 2250 screen, the programmed function keyboard, and the alphameric keyboard.

The first message on the screen, given by the GMS monitor, states that the user must depress any one of the programmed function keys.

The second message lists several options. The user should first type $NAMES, then depress the "Altn Coding" key and while it is held down, depress the "5" key on the alphameric keyboard. The list of programs should be examined carefully to ascertain that the QUANTAL program is listed. Next, one should type $LINK QUANTAL, followed by the "Altn Coding" and "5" key.

Once the QUANTAL program has been successfully obtained, a message will be displayed and the user, if entering for the first time, should depress programmed function key 1. If the user is entering the program following a departure to the CALCG program (see below), he should depress programmed function key 2 to return to his point of departure.

Next, the general instructions of the program will be listed. These instructions are important. They state:

1. At any point in the program you may restart by depressing programmed function key 30 or terminate by depressing programmed function key 31.
2. Your answers should be typed from the typewriter keyboard directly in front of you (unless you are asked differently).
3. All numbers are to be typed as real numbers -- decimal point must follow the number.
4. After each statement you should cause an "End of Block (EOB)" signal, by first depressing the "Altn Coding" key and, while it is held down, depressing the "5" key.
5. If there is more than one blank to be filled in a line, depress the "JUMP" key located on the left side of the typewriter keyboard to proceed from blank to blank.

The programmed function key 1, employed throughout the program to relate the normal flow of the program, should be depressed when the user is ready to continue.

Now the user will be asked to enter his first piece of data: the number of classes or batches on which he wishes to perform the analysis. This number must be greater than zero and less than one hundred. The number of classes or batches will ordinarily range between 5 and 25. With fewer than 4 classes the results are probably meaningless; more than 25 classes would be meaningful only if the total sample were very large. In the event that a number is entered which is not in the required range, an error message will be displayed and the user must re-enter a correct

number. The user must note that each number must be entered as a real number (with a decimal point actually typed, even if the number is an integer).

The next sequence of statements will request information on the concomitant variable or x-dosage which was administered to each batch, the number of individuals or subjects in each batch, and the number of subjects in each batch which responded. These three values will be filled into the blanks of the sentence: In batch xx where the concomitant variable (dosage) _________ was given to _____ subjects, there were _____ subjects that responded (successes). The user may proceed from blank to blank by depressing the "JUMP" key. The "End of Block" should be caused only when the entire sentence has been completed. In the first blank, the x-dosage (the mean x-score) for each batch may be entered as a real number in F or E (scientific notation) field specification. The width of the field is 14, thus data may be entered with maximums F14.7 or E14.7. If E format is used, the exponent (decimal point shift) must have two digits. For example, $1.62 * 10^{-8}$ would be entered as 1.62E-08. The x-dosage may be either negative or positive. However, since one of the important aspects of this program is to permit transformations on x - in order to assure better fit - the dosages should, as a rule, be positive (log and square-root are frequently used transformations).

In the second blank, the number of subjects or individuals in each batch is requested. It must be greater than two and must be entered as a real number.

The number of subjects which responded in each class occupies the third blank. It must be zero or greater and must not exceed the number of individuals in each batch.

If any of these requirements are not fulfilled, an error message will be displayed and the user must complete the statement again.

When the last statement of input has been successfully entered, the data will be displayed in an array. The user should examine his data carefully to ascertain that all data have been entered correctly. The user will then be given the opportunity to change data of any of the batches, to eliminate a batch, to add a new batch, or to continue with the analysis.

If the user wishes to correct any of the batches (more than one batch may be altered), he should depress programmed function key 2. At this time a message will appear that asks which batch the user desires to alter. If the number typed in by the user is not in the correct range, an error message will appear and the user must state the correct batch number. When this number has been entered satisfactorily, a statement to obtain input will be displayed on the screen. The user must complete this statement just as when he entered it the first time. However, he may eliminate this batch by simply refusing to fill in any of the blanks (responding EOB at once).

If the user wishes to add another batch to his data, he should depress programmed function key 5. It will then be necessary to complete the input statement according to the requirements pertaining to it.

After an addition or alteration has been made by the user, the entire data will again be displayed -- including corrections and addi-

tions -- and the user has the option once again to add or alter a batch or to continue.

When the data completely satisfy the user, he should depress programmed function key 1. At this time the analysis will be initiated and the user must decide which transformation he desires for his data. He may choose the PROBIT, LOGIT, LOG-LOG, ARC SIN, or WEIBULL by depressing programmed function key 6, 7, 8, 9, or 10 respectively. In the WEIBULL case he will also need to supply the exponent by depressing the appropriate key.

At this stage in QUANTAL, an alarm will be sounded and the screen will be blank for a few seconds. It is somewhat unpredictable how long this will take. The GMS monitor is time-shared. If the processor queue is very long, a minute or two may pass. If there is little batch activity, a few seconds may be all that is needed.

The first display of the output section will be the graph of the observed and predicted proportion of successes for each class. The x-dosages have been arranged in ascending order and are depicted on the horizontal axis. The batch numbers are written directly below this axis and the minimum and maximum x-dosage levels are written below the respective batch numbers. The vertical axis has the proportion of successes. The lined histogram is the predicted proportion of successes and the observed proportions of success are denoted by asterisks. The user should depress programmed function key 1 when he wishes to continue.

The next display (or series of displays) contains numerical answers (ten batches per display). On the final "page," the user may return to

"page 1" by depressing programmed function key 4. Displayed for each of the batches is the size of the batch, P(OBS) -- the observed proportion of successes, P(PRED) -- the predicted proportion of successes, Y(PRED) -- the predicted y-scores, Y(OBS) -- the observed y-scores, Y(WORK) -- the "working" quantities, the weights, X-DOSAGE -- the observed x-scores, and Y-ERROR -- the error of prediction in the y's.

The next display in the output section is an analysis of variance table. The sum-of-squares, degrees of freedom, and mean-squares are given for the regression and error. Also the calculated F = mean-squares regression/mean-squares error is displayed. The user then has the option to leave the QUANTAL in order to call the CALCG program, e.g., to find whether or not his F value is significant.

Upon returning to the QUANTAL program, if the user chose to call the CALCG program, he is reminded to return to this point by depressing programmed function key 2.

If the user does not wish to leave the QUANTAL program, he should depress programmed function key 1 when he wishes to continue.

Next there is a graph of the regression line $Y = \hat{\alpha} + \hat{\beta}x$ where $\hat{\alpha}$ and $\hat{\beta}$ have been estimated by the appropriate quantal analysis, and x is the dosage. The horizontal axis is the dosage-axis and the vertical axis is the y-axis. The observed y-values are denoted on the graph by asterisks. Programmed function key 1 is depressed to continue.

In the next section the user may alter his x-dosage by one of the following functions: sine, cosine, tangent, arc sine, arc cosine, arc tangent, exponential, natural logarithm, common logarithm, square root, absolute value and x-squared. To use one of the functions, he

depresses programmed function key 3 and then types in the number corresponding to the function desired. If a function is chosen which is incompatible with the data, e.g., square-root and negative x-dosages, an error message is displayed and the user must give correct answers.

If a function is chosen to alter the x-dosages, then the next section will be a display of the final (regression line) section with the x-scale altered. In fact the line itself will not be displayed -- only the asterisks. If the user likes this new plot he may perform reanalysis by continuing (depressing programmed function key 1). If he does not like it he will be given a chance, in the next display, to call the inverse of the first function and then, after yet another return, try a different function.

At the stage where the user is asked to supply the transformation he has an option to depress programmed function key 1 and thus obtain a new analysis (starting with the Summary) of the data in accordance with the last chosen transformation.

ILLUSTRATION

The QUANTAL program is called by the user and the instructions as outlined in the User's Guide are followed.

The data for the analysis are typed in by the user by his completing the sentences as shown in figure 3.1.

When all data have been entered, a tabular display of the data is exhibited (figure 3.2).

In this illustration, the probit transformation has been chosen by the user's depressing programmed function key 6.

After a short delay, for the calculations, there appears, on the screen, a step function indicating the estimated cumulative distribution function (proportion vs. dosage). The asterisks show the observed values. If the fit were perfect, they would appear coincident with the upper corner of each step. Figure 3.3 shows this display.

Next there appears a detailed numerical display (figure 3.4). Notice here that the greatest y-error is -.25310 in batch 4.

The next display (figure 3.5) is an analysis of variance table.

Next, the graph of the straight line $Y = \alpha + \beta x$ (as shown in figure 3.6) is displayed. It will be noted here that the analysis produces a very good fit for all but one of the data points.

In the second stage of this presentation, the user does not alter the x-dosages but chooses the logit transformation.

Figures 3.7 and 3.8 display the numerical results and the analysis of variance for the logistic transformation. As a comparison of the closeness of fit one could calculate R^2 = sum-of-squares regression/ sum-of-squares Total which is .9881 for the Probit and .9866 for the

logistic. This is an example of the (usual) closeness of the normal and logistic transformations. Note that no other base for comparison would be possible, since the dispersion of probits and logits are different (.975 corresponds to a probit of 1.96 (6.96) but to a logit of 3.66).

FIGURE 3.1

DATA INPUT STATEMENTS

```
     NEXT COMPLETE THE FOLLOWING STATEMENT FOR EACH OF THE    8
BATCHES.  AN EXAMPLE IS, IN BATCH 10. WHERE THE CONCOMITANT VARIABLE
(DOSAGE)     6.9     WAS GIVEN TO 36. SUBJECTS, THERE WERE 30. SUBJECTS
THAT RESPONDED (SUCCESSES).

     THE NUMBER OF SUBJECTS IN EACH BATCH MUST BE GREATER THAN 2.

     IN BATCH     1 WHERE THE CONCOMITANT VARIABLE (DOSAGE) 1.
WAS GIVEN TO 50.  SUBJECTS, THERE WERE  2.    SUBJECTS THAT RESPONDED
(SUCCESSES).

     IN BATCH     2 WHERE THE CONCOMITANT VARIABLE (DOSAGE) 1.5
WAS GIVEN TO 50.  SUBJECTS, THERE WERE  5.    SUBJECTS THAT RESPONDED
(SUCCESSES).

     IN BATCH     3 WHERE THE CONCOMITANT VARIABLE (DOSAGE) 2.
WAS GIVEN TO 40.  SUBJECTS, THERE WERE  7.    SUBJECTS THAT RESPONDED
(SUCCESSES).

     IN BATCH     4 WHERE THE CONCOMITANT VARIABLE (DOSAGE) 3.
WAS GIVEN TO 30.  SUBJECTS, THERE WERE  10.   SUBJECTS THAT RESPONDED
(SUCCESSES).

     IN BATCH     5 WHERE THE CONCOMITANT VARIABLE (DOSAGE) 3.5
WAS GIVEN TO 20.  SUBJECTS, THERE WERE  12.   SUBJECTS THAT RESPONDED
(SUCCESSES).

     IN BATCH     6 WHERE THE CONCOMITANT VARIABLE (DOSAGE) 4.
WAS GIVEN TO 40.  SUBJECTS, THERE WERE  30.   SUBJECTS THAT RESPONDED
(SUCCESSES).

     IN BATCH     7 WHERE THE CONCOMITANT VARIABLE (DOSAGE) 4.5
WAS GIVEN TO 30.  SUBJECTS, THERE WERE  26.   SUBJECTS THAT RESPONDED
(SUCCESSES).

     IN BATCH     8 WHERE THE CONCOMITANT VARIABLE (DOSAGE) 5.5
WAS GIVEN TO 50.  SUBJECTS, THERE WERE  43.   SUBJECTS THAT RESPONDED
(SUCCESSES).
```

FIGURE 3.2

TABULAR DISPLAY OF DATA

YOUR DATA NOW IS:

BATCH NUMBER	DOSAGE	NUMBER OF SUBJECTS	NUMBER OF SUCCESSES
1	1.00000	50	2
2	1.50000	50	5
3	2.00000	40	7
4	3.00000	30	10
5	3.50000	20	12
6	4.00000	40	30
7	4.50000	30	26
8	5.50000	50	43

IF THERE IS ANY CORRECTION TO YOUR DATA, DEPRESS PROGRAM FUNCTION KEY 2. IF YOU WISH TO ADD ANOTHER CLASS (BATCH) DEPRESS PROGRAM FUNCTION KEY 5. IF YOU WISH TO CONTINUE, DEPRESS PROGRAM FUNCTION KEY 1.

FIGURE 3.3

THE ESTIMATED CUMULATIVE DISTRIBUTION

THE GRAPH BELOW DISPLAYS THE OBSERVED AND PREDICTED PROPORTION OF SUCCESS FOR EACH BATCH (CLASS).

THE HORIZONTAL AXIS IS THE X-DOSAGE OF EACH BATCH (ARRANGED IN ASCENDING ORDER). THE BATCH NUMBER FOR EACH CORRESPONDING BATCH IS DIRECTLY UNDER THE AXIS AND THE TWO LOWER FIGURES ARE THE MINIMUM AND MAXIMUM X-DOSAGE.

THE VERTICAL AXIS IS THE PROPORTION OF SUCCESSES AXIS. THE LINED GRAPH IS THE PREDICTED PROPORTION OF SUCCESSES AND THE ASTERICKS DENOTE THE OBSERVED PROPORTION OF SUCCESSES.

WHEN YOU ARE READY TO PROCEED, DEPRESS PROGRAM FUNCTION KEY 1.

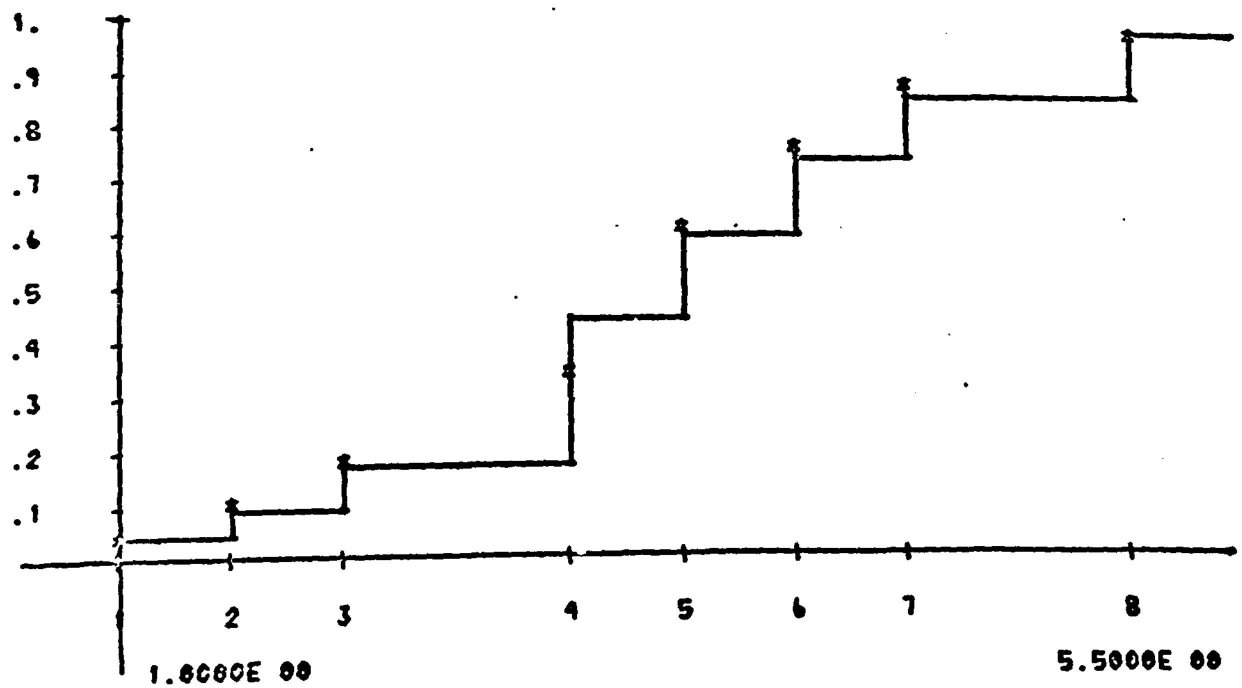


FIGURE 3.4

NUMERICAL RESULTS OF THE ANALYSIS

PROBIT ANALYSIS

THIS IS THE LAST PAGE OF THIS SECTION OF YOUR OUTPUT.
IF YOU WISH TO REVIEW THIS SECTION, DEPRESS PROGRAM FUNCTION KEY 4
OTHERWISE, DEPRESS PROGRAM FUNCTION KEY 1.

CLASS	1	2	3	4	5
SIZE	50	50	40	30	20
P(OBS)	4.0000E-02	1.0000E-01	1.7500E-01	3.3333E-01	6.0000E-01
P(PRED)	3.9746E-02	8.6990E-02	1.6717E-01	4.2960E-01	5.8507E-01
Y(PRED)	-1.7537E 00	-1.3595E 00	-9.6542E-01	-1.7719E-01	2.1693E-01
Y(OBS)	-1.7511E 00	-1.2817E 00	-9.3480E-01	-4.3069E-01	2.5335E-01
Y(WORK)	-1.7507E 00	-1.2774E 00	-9.3413E-01	-4.2282E-01	2.5320E-01
WEIGHTS	9.6282E 00	1.5781E 01	1.8005E 01	1.8802E 01	1.2516E 01
XDOSAGE	1.0000E 00	1.5000E 00	2.0000E 00	3.0000E 00	3.5000E 00
Y-ERROR	2.5206E-03	7.7810E-02	3.0919E-02	-2.5310E-01	3.6419E-02

CLASS	6	7	8
SIZE	40	30	50
P(OBS)	7.5000E-01	8.6667E-01	9.6000E-01
P(PRED)	7.2941E-01	8.4259E-01	9.6354E-01
Y(PRED)	6.1105E-01	1.0052E 00	1.7934E 00
Y(OBS)	6.7449E-01	1.1103E 00	1.7507E 00
Y(WORK)	6.7323E-01	1.1052E 00	1.7490E 00
WEIGHTS	2.2205E 01	1.3107E 01	9.0858E 00
XDOSAGE	4.0000E 00	4.5000E 00	5.5000E 00
Y-ERROR	6.3445E-02	1.0561E-01	-4.2710E-02

FIGURE 3.5

ANALYSIS OF VARIANCE TABLE

PROBIT ANALYSIS

ANALYSIS OF VARIANCE TABLE

SOURCE	SUM OF SQUARES	DEGREES OF FREEDOM	MEAN SQUARE
REGRESSION	0.12601057E 03	1	0.12601057E 03
ERROR	0.15121155E 01	6	0.25201923E 00
TOTAL	0.12752269E 03	7	

CALCULATED F = 0.50000366E 03

IF YOU WISH TO LEAVE THE QUANTAL PROGRAM TO DETERMINE F(1, 6),BY CALLING THE "CALCG" PROGRAM, DEPRESS PROGRAM FUNCTION KEY 11. IF YOU WISH TO CONTINUE, DEPRESS PROGRAM FUNCTION KEY 1.

FIGURE 3.6

GRAPHIC DISPLAY

THE FOLLOWING IS THE GRAPH OF THE STRAIGHT LINE Y = ALPHA + BETA * X
WHERE ALPHA AND BETA HAVE BEEN ESTIMATED IN THE PROBIT ANALYSIS.
HERE THE EQUATION IS Y = -0.259139E 01 + 0.768234E 00 * X.
THE HORIZONTAL AXIS IS THE DOSAGE-AXIS AND THE VERTICAL AXIS IS THE Y-AXIS

THE VALUES DENOTED BY '*' ARE THE OBSERVED Y-VALUES

WHEN YOU ARE READY TO PROCEED, DEPRESS PROGRAM FUNCTION KEY 1.

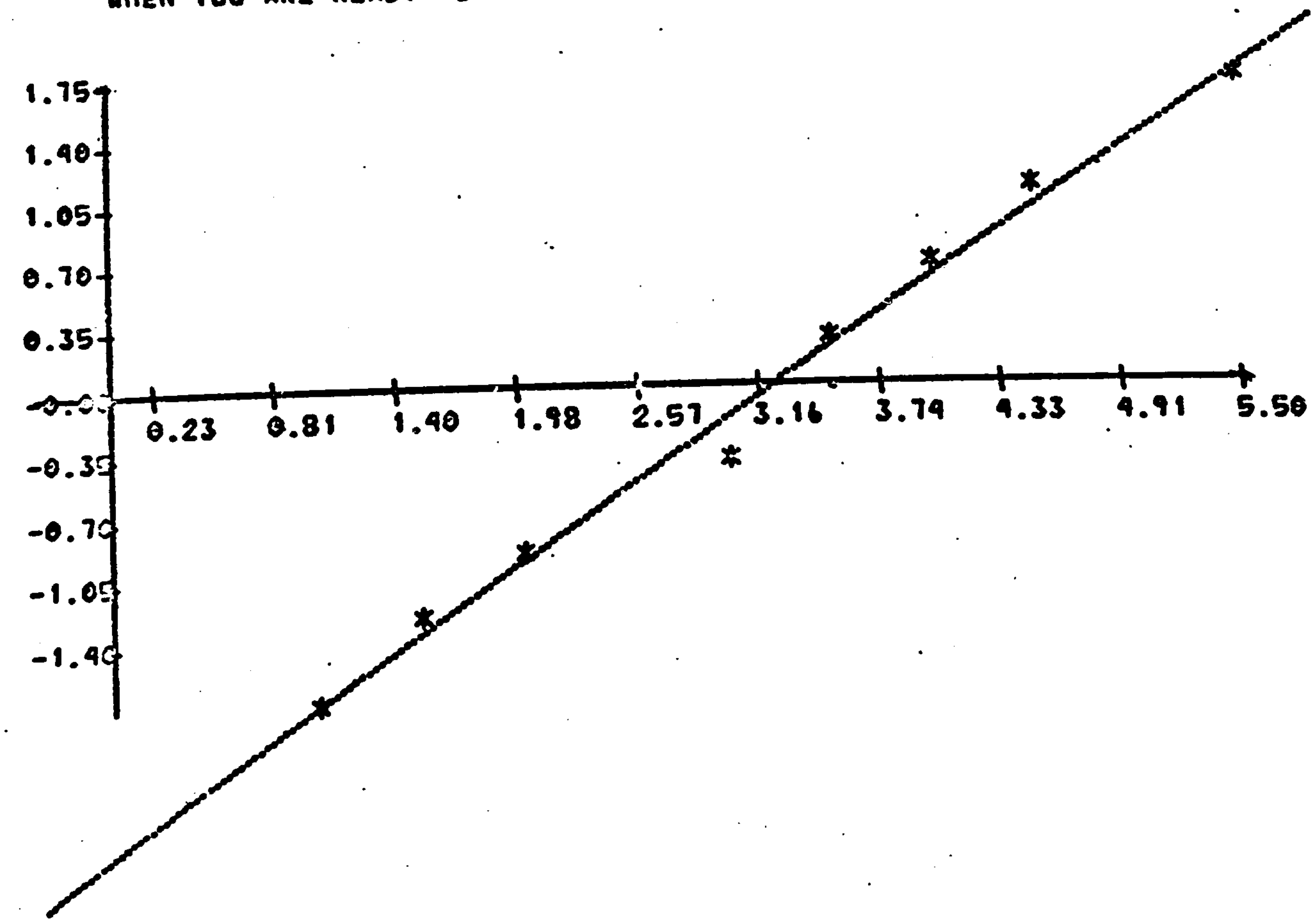


FIGURE 3.7

NUMERICAL RESULTS OF THE ANALYSIS

LOGIT ANALYSIS

THIS IS THE LAST PAGE OF THIS SECTION OF YOUR OUTPUT.
IF YOU WISH TO REVIEW THIS SECTION, DEPRESS PROGRAM FUNCTION KEY 4
OTHERWISE, DEPRESS PROGRAM FUNCTION KEY 1.

CLASS	1	2	3	4	5
SIZE	50	50	40	30	20
P(OBS)	4.0000E-02	1.0000E-01	1.7500E-01	3.3333E-01	6.0000E-01
P(PRED)	4.4293E-02	8.4340E-02	1.5473E-01	4.1963E-01	5.8965E-01
Y(PRED)	-3.0716E 00	-2.3[illegible]E 00	-1.69[illegible]E 00	-3.2[illegible]E-01	3.6253E-01
Y(OBS)	-3.1781E 00	-2.1972E 00	-1.[illegible]E 00	-6.931[illegible]E-01	4.0546E-01
Y(WORK)	-3.1730E 00	-2.1820E 00	-1.5430E 00	-6.7[illegible]E-01	4.6329E-01
WEIGHTS	2.1166E 00	3.8613E 00	5.2316E 00	7.3062E 00	4.8392E 00
XDOSAGE	1.0000E 00	1.5000E 00	2.0000E 00	3.0000E 00	3.5000E 00
Y-ERROR	-1.0643E-01	1.0759E-01	1.4736E-01	-3.6[illegible]E-01	4.2931E-02

CLASS	6	7	8
SIZE	40	30	50
P(OBS)	7.5000E-01	8.6667E-01	9.6000E-01
P(PRED)	7.4068E-01	8.[illegible]E-01	9.[illegible]E-01
Y(PRED)	1.0494E 00	1.73[illegible]E 00	3.1[illegible]E 00
Y(OBS)	1.[illegible]E 00	1.[illegible]E 00	[illegible]
Y(WORK)	1.[illegible]E 00	1.[illegible]E 00	[illegible]
WEIGHTS	7.6334E 00	3.[illegible]E 00	2.04[illegible]E 00
XDOSAGE	4.0000E 00	4.5000E 00	5.0000E 00
Y-ERROR	4.9247E-02	1.3[illegible]E-01	6.0[illegible]E-02

NOT REPRODUCIBLE

FIGURE 3.8

ANALYSIS OF VARIANCE TABLE

LOGIT ANALYSIS

ANALYSIS OF VARIANCE TABLE

SOURCE	SUM OF SQUARES	DEGREES OF FREEDOM	MEAN SQUARE
REGRESSION	0.97694717E 02	1	0.97694717E 02
ERROR	0.13233643E 01	6	0.22056067E 00
TOTAL	0.99018082E 02	7	

CALCULATED F = 0.44293799E 03

IF YOU WISH TO LEAVE THE QUANTAL PROGRAM TO DETERMINE F(1, 6),BY CALLING THE "CALCG" PROGRAM, DEPRESS PROGRAM FUNCTION KEY 11. IF YOU WISH TO CONTINUE, DEPRESS PROGRAM FUNCTION KEY 1.

CHAPTER IV

PROGRAM DOCUMENTATION

The QUANTAL program has been written in FORTRAN IV. The listing is presented in Appendix A. Inasmuch as execution of the program must use no more than 64 K bytes of core (16 K words), an overlay structure is required. The exposition (figure 4.1) shows the job control cards for this overlay.

As may be seen in figure 4.2, the overlay structure for the QUANTAL program is relatively simple. The controlling segment, or root segment contains the MAIN program and several basic GRAF (Graphics Addition to FORTRAN) routines. This segment remains in storage throughout execution of the QUANTAL program and branches off into two separate segments. One of the segments contains the FORTRAN routines that are necessary for the mathematical calculations of the program and also the subroutines CALCG and FNS. The other branch of QUANTAL contains the GRAF routines and the subroutines which, through use of these routines, obtain the data from the user and display the results of the analysis. The three subroutines, OUTPUT, CHNGTR, and GRACDF, require the GRAF routines, CORDCALL, LINE$$, SIZE, $CORD$, and POINT$, to present various graphical displays. Only one subroutine, with the necessary FORTRAN or GRAF routines provided by the structure resides in storage with the root segment at any one time and thus the total size of the program is appropriately restricted.

THE MAIN PROGRAM

The reader is referred to reference [7] for an explanation of the meaning of each of the GRAF (Graphics Addition to FORTRAN) subroutines.

The main program of the QUANTAL program is designed to call various subroutines, in an order primarily designated by the user. The flow of the program is determined by the value of KEY, a fixed point value which is stored in COMMON.

VALUE OF KEY	COMMENT
1	Restart the program
2	Proceed to display data
3	Proceed to choose transformation
4	Proceed to change function
5	Proceed to display data
6	Proceed to alter x-dosage
7	Same as 3
10	Alter one class of data
50	Terminate the program

The COMMON set

NDET: An array required by GRAF

X(100): Reserved for x-dosage

N(100): Number of subjects per batch

NR(100): Number of successes per batch

DV1 to DVB inclusive: DV (display-variable) names required by GRAF

NCL: Number of batches

KEY: Control of program flow (see above)

KK: Designation for batch to be changed

IFUN: Indicates function on x-dosage

ITRAN: Indicates tranformation on P (Probit, Logistic, etc.)

IREC: Index for random access I/O.

SUBROUTINE INPUTA

Purpose: The purpose of subroutine INPUTA is to obtain the number of classes in the analysis.

Procedure:

STATEMENT NUMBER	FORTRAN STATEMENT NUMBER	COMMENT
3		The Graphics Data Control Block (GDCB) is opened and the display variables DV1, DV2, DV3, DV4, and DVE are established, thereby initializing the 2250 for I/O.
4		The lights of programmed function keys 1-11 and 30 and 31 are turned on.
6-8		The scope and display variables are cleared from previous commands.
9-12		The message in FORMAT 400 is plotted on the screen to enable the user, if returning from the CALCG program to proceed directly to his point of departure in the QUANTAL program.
13	402	Procedure is detained until the user responds.
14		If programmed function key 2 has been depressed, (NDET(4) equals 2), indicating a return from the CALCG program, proceed to FORTRAN statement 401.
15		If programmed function key 1 has been depressed, proceed to FORTRAN statement 66.
16		If neither of these programmed function keys has been depressed, return to statement 402

SUBROUTINE INPUTA (Continued)

STATEMENT NUMBER	FORTRAN STATEMENT NUMBER	COMMENT
		(await proper reply).
17	66	The scope is cleared of all existing displays.
18		Display variable DV1 is reinitialized.
19-25		The messages in FORMAT statements 100 and 1000, the general instructions of the program, are displayed on the screen.
26	201	Procedure is detained until user has finished reading the display.
27		If programmed function key 31 has been depressed (NDET(4) equals 31), indicating the user wishes to terminate the program, proceed to FORTRAN statement 750 (go out).
28		If programmed function key 30 has been depressed (NDFT(4) equals 30), indicating the user wishes to restart the program, proceed to FORTRAN statement 67.
29		If an interrupt has been given by the user from any other device except the programmed function keyboard (I is not equal to 1), return to detain statement 201.
30		The scope is cleared of all existing displays.
31-34	25	The message in FORMAT statement 101, asking the user to type in the number of batches in the analysis, is plotted on the screen.

SUBROUTINE INPUTA (Continued)

STATEMENT NUMBER	FORTRAN STATEMENT NUMBER	COMMENT
35		A cursor is placed in the first unprotected position of FORMAT statement 101 so that the user can type in the number of batches.
36	200	Procedure is detained until user types in number of batches.
37		If programmed function key 31 has been depressed, indicating the user wishes to terminate the program, proceed to FORTRAN statement 750.
38		If programmed function key 30 has been depressed, indicating the user wishes to restart the program, proceed to FORTRAN statement 67.
39	69	If an interrupt has been given by the user from any device except the alphameric keyboard (I is not equal to 2) return to detain statement 200.
40-43		XNUM is NCL, floated.
44		The pointers for dummy unit 4 are reset.
45-47		The cursor is removed from the screen.
48		The integer variable NCL is equal to XNUM.
49		If the number of classes (NCL) is less than zero or greater than 100, proceed to FORTRAN statement 30 where an error message is displayed.
50	50	When the statement has been completed correctly all error messages are erased.

SUBROUTINE INPUTA (Continued)

STATEMENT NUMBER	FORTRAN STATEMENT NUMBER	COMMENT
52-55	30	The error message in FORMAT statement 114 is displayed on the screen.
56-57		The incorrect statement 101 is removed from the Buffer Table and display variable DV2 is re-initialized.
58		Display variable DVE is re-initialized.
60	401	KEY is set equal to 25.
61		The screen is cleared of all existing displays.
64	21	Display variables DV1, DV2, DV3, DV4, and DVE are re-initialized.

SUBROUTINE DATA

Purpose: The purpose of this subroutine is to obtain the data from the user by his completing the sentence: In batch XX where the concomitant variable (dosage) _________ was given to _____ subjects, there were _____ subjects that responded (successes).

Procedure:

STATEMENT NUMBER	FORTRAN STATEMENT NUMBER	COMMENT
3		The display variables DV3, DV4, and DVE are established.
4		The scope is cleared of all existing displays.
5		If KEY is equal to 10, indicating data is to be obtained from only one batch (when changes are made), proceed to FORTRAN statement 75.
6		NL, the number of times FORMAT statement 103 is to be written is initialized equal to NCL.
7		JEY is a flag = 0 for all but the last page, = 1 for the last page.
8		NI, the increment of loop 500 is initialized.
9-12	4	The message in FORMAT 102 to obtain data from a batch is plotted on the screen.
13-19		Only eight statements are to appear on the screen at a time. NN is the delimiter of loop 500.
20-23		The statement to obtain data from the user is plotted on the screen.

SUBROUTINE DATA (Continued)

STATEMENT NUMBER	FORTRAN STATEMENT NUMBER	COMMENT
24		A cursor is placed in the first unprotected position of FORMAT statement 103 so that the user can type in data.
25	201	Procedure is detained until the user has entered data for a batch.
26		If programmed function key (PFK) 31 has been depressed (NDET(4) equals 31), indicating the user wishes to terminate the program, proceed to FORTRAN statement 750 (exit).
27		If PFK 30 has been depressed (NDET(4) equals 30), indicating the user wishes to begin the program again, proceed to FORTRAN statement 67.
28		If an interrupt has been given by the user from any device except the alphameric keyboard (I is not equal to 2) return to detain statement 201.
29-32		X(KK), the x-dosage; and XX and XY, the real values for the number of individuals in batch KK and the number of successes in batch KK, are read according to FORMAT 301.
33		The pointers for dummy unit 4 are reset.
34		N(KK) is defined equal to XX.
35		NR(KK) is defined equal to XY.

SUBROUTINE DATA (Continued)

STATEMENT NUMBER	FORTRAN STATEMENT NUMBER	COMMENT
37	22	If the number of individuals in each batch is less than 3, proceed to FORTRAN statement 11 to write an error message.
38	12	If the number of individuals who responded is negative, proceed to FORTRAN statement 13 to write an error message. If the number of individuals is positive, proceed to next statement.
39	15	If the number of individuals who responded is greater than the number of individuals in that batch, proceed to FORTRAN statement 45 to write an error message.
41-43	20-21	The beam is placed in the upper left-hand corner of the screen.
45-48	11	The error message in FORMAT statement 117 is displayed on the screen.
49-50		The incorrect statement 103 is removed from the Buffer Table and display variables DV4 and DVE are re-initialized.
52-55	13	The error message in FORMAT statement 118 is plotted on the screen.
56-57		The incorrect statement 103 is removed from the Buffer Table and display variables DV4 and DVE are re-initialized.

SUBROUTINE DATA (Continued)

STATEMENT NUMBER	FORTRAN STATEMENT NUMBER	COMMENT
59-62	45	The error message in FORMAT statement 120 is displayed on the screen.
63		Incorrect statement 103 is removed from the BT.
64		The cursor is removed from the screen.
65		Display variables DV4 and DVE are re-initialized.
67-69	501	The cursor is removed from the screen.
70		Display variable DV4 is re-initialized.
71	500	End of loop 500 - All error messages are erased from the screen.
72		The scope is cleared of all existing displays.
73-74		NI and NL - the increment and decrement of loop 500 are up-dated.
75		If JEY is equal to 1, indicating all data has been obtained, exit; otherwise, proceed to FORTRAN statement 4.
76-77	75	When subroutine DATA has been called with KEY=10, the increment and delimiter of loop 500 are set to the desired batch KK.
78		JEY is set equal to 1.
79-82		FORMAT statement 192 is displayed on the screen.
84	750	KEY is set equal to 50.
86	67	KEY is set equal to 1.
87	99	Display variables DV3, DV4, and DVE are re-initialized.

SUBROUTINE INPUTB

Purpose: The purpose of this subroutine is to display the summary of the data obtained in subroutine DATA and to allow the user to alter any data or to add or remove one or more classes to the analysis.

Procedure:

STATEMENT NUMBER	FORTRAN STATEMENT NUMBER	COMMENT
3		The display variables DV5, DV6, DV7, DVD, DVA, and DVE are established.
4	502	The scope is cleared of all existing displays.
5-7		The title message, FORMAT statement 104, is written onto display variable DV5.
8		K, the number of batches to be displayed on the screen at one time, is initialized equal to 0.
9		Loop 21 - One class of individuals at a time, and the corresponding x-value, N-value, and NR-value are to be plotted on the screen at a time.
10-13		The number of the class, the x-value, N-value, and NR-value are plotted on the screen according to FORMAT statement 105.
14		Display variables DV5, and DV7 are re-initialized.
15		K is incremented by one.
16		A statement to insure that no more than 25 rows appear on a single display.

SUBROUTINE INPUTB (Continued)

STATEMENT NUMBER	FORTRAN STATEMENT NUMBER	COMMENT
17		The screen is cleared of all existing displays.
18		K is re-initialized equal to 0.
19-20		The title message, FORMAT statement 104, is written onto display variable DV5.
22-25		A message which gives the user the ability to correct any data or to add another class to his analysis is displayed on the screen.
26	202	Procedure is detained until user decides whether or not he wishes to change data.
27		If an interrupt has been given by the user from any device except the programmed function keyboard (J is not equal to 1) return to detain statement 202.
28	7	All error messages are cleared from the screen.
29		If programmed function key (PFK) 31 has been depressed (NDET(4) equals 31), indicating the user wishes to terminate the program, proceed to FORTRAN statement 750 (exit).
30	8	If PFK 30 has been depressed (NDET(4) equals 30), indicating the user wishes to restart the program, proceed to FORTRAN statement 67.
31		If PFK 1 has been depressed (NDET(4) equals 1), indicating the user does not wish to alter any data, proceed to FORTRAN statement 76.

SUBROUTINE INPUTB (Continued)

STATEMENT NUMBER	FORTRAN STATEMENT NUMBER	COMMENT
32		If PFK 2 has been depressed (NDET(4) equals 2), indicating the user wishes to make a correction to his data, proceed to FORTRAN statement 68.
33		If PFK 5 has been depressed (NDET(4) equals 5), indicating the user wishes to add another batch to his analysis, proceed to FORTRAN statement 300.
34		If none of PFK 31,30, 1, 2, or 5 have been depressed, return to detain statement 202.
35	300	The number of classes in the analysis, NCL, is increased by one (a batch is to be added).
36		The scope is cleared of all existing displays.
37		KK, the number of the batch to be added is set equal to NCL.
38		The variable KEY is set equal to 10, to indicate subroutine DATA should be called again.
40	68	The scope is cleared of all existing displays.
41		Display variable DVA is re-initialized.
42-45		The message in FORMAT statement 203, which determines in which batch the user wishes to change the data, is displayed on the screen.
46		A cursor is placed in the first unprotected position of FORMAT statement 203 so that the user can type in the batch number to be corrected.

SUBROUTINE INPUTB (Continued)

STATEMENT NUMBER	FORTRAN STATEMENT NUMBER	COMMENT
47	206	Procedure is detained until the user decides which batch he wants to change.
48		If PFK 31 has been depressed (NDET(4) equals 31), indicating the user wishes to terminate the program, proceed to FORTRAN statement 750.
49		If PFK 30 has been depressed (NDET(4) equals 30), indicating the user wishes to restart the program, proceed to FORTRAN statement 67.
50		If an interrupt has been given by any other device than the alphameric keyboard (I does not equal 2), return to detain statement 206.
51-54		The floated value XK of KK is read according to FORMAT statement 207.
55		The pointers for dummy unit 4 are reset.
56		KK is defined equal to XK.
57		If the number of the batch which is to be corrected is less than 0 or greater than NCL, proceed to FORTRAN statement 209 to write an error message.
58-60		The cursor is removed from the screen.
61		KEY is set equal to 10, to indicate subroutine DATA should be called again.
63-66	209	The error message in FORMAT statement 211 is displayed on the screen.

SUBROUTINE INPUTB (Continued)

STATEMENT NUMBER	FORTRAN STATEMENT NUMBER	COMMENT
67		Incorrect statement 203 is cleared from the screen.
68		Display variable DVE is re-initialized.
70	750	KEY is equated to 50.
72	76	KEY is equated to 7.
74	67	KEY is equated to 1.
75	74	The screen is cleared of all existing displays.
76		Display variables DV1, DV2, DV3, DV4, DV5, DV6, DV7, DVD, and DVE are re-initialized.

SUBROUTINE CHNGTR

Purpose: Subroutine CHNGTR allows the user a choice between five transformations: the Probit, the Logistic, Log-Log, Arc Sin and Weibull.

Procedure:

STATEMENT NUMBER	FORTRAN STATEMENT NUMBER	COMMENT
3		Display variable DV1 is established.
4-7		FORMAT statement 1 is displayed on the screen.
8	2	Procedure is detained until the user has chosen the transformation.
9		If an interrupt has been given by the user from any device except the programmed function keyboard (J is not equal to 1) return to detain statement 2.
10		If programmed function key (PFK) 31 has been depressed (NDET(4) equals 31), indicating the user wishes to terminate the program, proceed to FORTRAN statement 750 (exit).
11		If PFK 30 has been depressed (NDET(4) equals 30), indicating the user wishes to restart the program, proceed to FORTRAN statement 67.
12		If PFK 6 has been depressed (NDET(4) equals 6), indicating the user desires the Probit transformation, ITRAN is set equal to 1.

SUBROUTINE CHNGTR (Continued)

STATEMENT NUMBER	FORTRAN STATEMENT NUMBER	COMMENT
13		If PFK 7 has been depressed (NDET(4) equals 7), indicating the user desires the Logit transformation, ITRAN is set equal to 2.
14		If PFK 8 has been depressed (NDET(4) equals 8), indicating the user desires the Log-Log transformation, ITRAN is set equal to 3.
15		If PFK 9 has been depressed (NDET(4) equals 9), indicating the user desires the Arc Sin transformation, ITRAN is set equal to 4.
16		If PFK 10 has been depressed (NDET(4) equals 10), indicating the user desires the Weibull transformation, ITRAN is set equal to 5.
17		If any other programmed function key has been depressed, return to detain statement 2.
19	750	KEY is set equal to 50
21	67	KEY is set equal to 1.
22	99	Display variable DV1 is re-initialized.
23		The screen is cleared of all existing displays.

SUBROUTINE CALC

Purpose: This is a conventional maximum-likelihood routine for the quantal analysis.

Output:

- P - a single-dimensioned field, in floating point, which contains the observed proportion of successes for each batch
- PCAP - a single-dimensioned field, in floating point, containing the predicted proportion of successes for each batch
- YPR - a single-dimensioned field, in floating point, containing the predicted y-scores for each batch
- Y - a single-dimensioned field, in floating point containing the observed y-scores for each batch
- YWORK - a single-dimensioned field, in floating point, containing the "working" quantities of each batch
- W - a single-dimensioned field, in floating point, containing the weights of each batch
- ER - a single-dimensioned field, in floating point, containing the error of prediction of the y-scores in each batch
- SSR - a floating point number which is the sum-of-squares for regression
- SSE - a floating point number which is the sum-of-squares for error
- ALF - a floating point number which estimates α
- BETA - a floating point number which estimates β

The output variables are stored on a direct access disk, logical unit 28.

Procedure:

STATEMENT NUMBER	FORTRAN STATEMENT NUMBER	COMMENT
5		KMAX is set equal to 0, indicating termination when alpha or beta converge*.
6		K, the number of the cycle, is initialized equal to 0.
7-8		ALFO and BETO, the initial estimates of ALF and BETA are initialized equal to 0.
9	101	K, the number of the cycle is tested. If K is negative, go out; if K is equal to zero, proceed to start; and if K is positive, proceed to FORTRAN statement 21 (iteration).
10-11	3	Loop 58 - The number of successes are translated into proportion of successes in each class.
12-19		Loop 5 - To avoid singularities, if a class has no successes, it will be translated as though it had ½ successes. If all individuals in some class show success, (n - ½) successes will be recorded.
20-32	77	Loop 11 - Y(Observed) is calculated according to the chosen transformation: for the Probit transformation, function YORMP is called; for the Logit - DFP; for the Log-Log - DFP3; for

*This subroutine was adopted from a batch-mode program which permitted other modes of termination.

SUBROUTINE CALC (Continued)

STATEMENT NUMBER	FORTRAN STATEMENT NUMBER	COMMENT
		the Arc Sine transformation - DFP4; and for the Weibull transformation, DFP2 is called.
33		K, the number of cycles is tested. If K is negative, go out; if K is equal to zero, proceed to FORTRAN statement 10; and if K is positive, proceed to FORTRAN statement 21.
34-36	10	Loop 12 - Initially the working y-scores are set equal to the observed y-scores and the weights are set equal to the n_i.
37		Bypass recalculation if this is initial loop.
38-52	21	Loop 23 - The Z-field is calculated according to the chosen transformation: for the Probit transformation, function YORMZ is used; for the Logit transformation, DFZ is used; for the Log-Log transformation, DFZ3 is used; for the Arc-Sin transformation, DFZ4 is used; and for the Weibull transformation, DFZ2 is used.
53-54	38	No function here (came from batch program).
55	41	If the cycle number is negative (error), go out; otherwise, proceed to FORTRAN statement 84.
56	84	The weighted regression subroutine, WREG, is called to estimate the predicted y-scores, alpha, and beta.

SUBROUTINE CALC (Continued)

STATEMENT NUMBER	FORTRAN STATEMENT NUMBER	COMMENT
57		The number of the cycle is increased by one.
58.		If ALF is relatively close to ALFO or BETA is relatively close to BETO, proceed to the next statement. Otherwise, proceed to FORTRAN statement 60.
60	60	ALFO is re-initialized equal to ALF.
61		BETO is re-initialized equal to BETA.
62		If K, the cycle number is less than 5, proceed to FORTRAN statement 141; otherwise, proceed to FORTRAN statement 99.
63-75	141	Loop 65 - PCAP, the predicted proportion of successes is calculated for each batch according to the chosen transformation: for the Probit transformation, YORMX is used; for the Logit transformation, DFX is used; for the Log-Log, DFX3 is used; for the Arc Sin, DRX4 is used; and for the Weibull transformation, DFX2 is used.
76		Make another iteration.
78-86	99	P, PCAP, YPR, Y, YWORK, W, ER, SSR, SSE, ALF, and BETA are written onto direct access disk 22
87	2001	Control is returned to the calling program.

PROBIT FUNCTIONS: YORMZ(Y)

YORMX(Y)

YORMP(P)

Definitions:

Z = YORMZ(Y) $\qquad Z = [1/(2\pi)^{\frac{1}{2}}]\ e^{-\frac{1}{2}y^2}$

P = YORMX(Y) $\qquad P = [1/(2\pi)^{\frac{1}{2}}] \int_{-\infty}^{y} e^{-\frac{1}{2}t^2}\, dt$

Y = YORMP(P) $\qquad$ obtains Y from P as defined in YORMX

Procedure:

YORMZ: trivial

YORMX: If $|Y| \leq 3.0$, YORMX is evaluated by the continued fraction expansion

$$\tfrac{1}{2} + [1/(2\pi)^{\frac{1}{2}}]\ e^{-\frac{1}{2}y^2}\ \left\{ \frac{y}{1-}\ \frac{y^2}{3+}\ \frac{2y^2}{5-}\ \frac{3y^2}{7+} \ldots \right\}$$

which is continued until the terms begin declining and the relative error is less than 10^{-14}.

If $|Y| > 3.0$, the continued fraction

$$1 - [1/(2\pi)^{\frac{1}{2}}]\ e^{-\frac{1}{2}y^2}\ \left\{ \frac{1}{y+}\ \frac{1}{y+}\ \frac{2}{y+}\ \frac{3}{y+} \ldots \right\}$$

is employed.

YORMP: As a first guess, the Hastings form for inverting the erfc-function is employed. This is improved by a Newton iteration, until the relative error (P calculated from YORMX - P input)/(P input) is less than 10^{-6}.

PROBIT FUNCTIONS: (Continued)

Limitations:

If $P \leq 0$, -DOMEG is returned; if $P \geq 1$, +DOMEG is returned, where DOMEG = $.99999999 \times 10^{38}$.

LOGIT FUNCTIONS: DFZ(Y)

DFX(Y)

DFP(P)

Definitions:

Z = DFZ(Y) $\qquad Z = e^y/(1 + e^y)^2$

P = DFX(Y) $\qquad P = e^y/(1 + e^y)$

Y = DFP(P) $\qquad Y = \log(P/(1 - P))$

LOG - LOG FUNCTIONS: DFZ3(Y)

DFX3(Y)

DFP3(P)

Definitions:

Z = DFZ3(Y) $\qquad Z = -P \log(P)$, where $P = e^{-e^{-Y}}$

P = DFX3(Y) $\qquad P = e^{-e^{-Y}}$

Y = DFP3(P) $\qquad Y = -\log(-\log(P))$

ARC SIN FUNCTIONS: DFZ4(Y)

DFX4(Y)

DFP4(P)

Definitions:

Z = DFZ4(Y) $\qquad Z = \frac{1}{2} \cos(Y)$

P = DFX4(Y) $\qquad P = \frac{1}{2}(1 + \sin(Y))$

Y = DFP4(P) $\qquad Y = \text{arsin}(2P - 1)$

WEIBULL FUNCTIONS: DRZ2(Y)

DFX2(Y)

DFP2(P)

Definitions:

Z = DFZ2(Y) $Z = Y^{r-1}e^{-Y^r/r}$

P = DFX2(Y) $P = 1 - e^{-Y^r/r}$

Y = DFP2(P) $Y = [-r \log(1 - P)]^{1/r}$

SUBROUTINE WREG:

This is a conventional weighted regression routine to obtain SSR, the sum-of-squares for regression; SSE, the sum-of-squares for error; VSE, the mean squares for error; ALF; BETA; and R^2, which is SS Reg/SS Total. The weighted regression routine is applied to the "working" quantities while the determination of error uses the observed quantities as input.

SUBROUTINE GRACDF

Purpose: The purpose of this subroutine is to display a graph of the observed and predicted proportion of successes for each batch (class).

Input:

P - a single-dimensioned field, read from direct access disk, containing the observed proportion of successes for each batch.

PCAP - a single-dimensioned field, read from direct access disk, containing the predicted proportion of successes for each batch

Procedure:

STATEMENT NUMBER	FORTRAN STATEMENT NUMBER	COMMENT
5		Display variable DVD is established.
6-7		P and PCAP are read from direct access disk 28.
8		Display variable DVD is re-initialized.
9		The scope is cleared of all existing displays.
10-14		Loop 10 stores the integers 1 to NCL in the single-dimensioned field IS, field X into single-dimensioned field XS, field P into single-dimensioned field PS, and field PCAP into single-dimensioned field PCAPS.
15-32		Loop 500 sorts XS into ascending order, rearranging the number of the batch (IS), the observed proportion of successes (PS) and the predicted

SUBROUTINE GRACDF (Continued)

STATEMENT NUMBER	FORTRAN STATEMENT NUMBER	COMMENT
		proportion of successes (PCAPS) in corresponding order.
33		XMIN, the minimum of the x-dosages is set equal to XS(1).
34		XMAX, the maximum of the x-values is set equal to XS(NCL).
35-36		XUP and XLOW are calculated for the upper and lower coordinates of the horizontal axis of the graph.
37-40		FORMAT statement 1000, an explanation of the graph, is plotted on the screen.
41		The coordinates of the lower left corner of the screen (XLOW, -.2) and the upper right corner (XUP, 2.) are established by the UCORD command.
42-43		The vertical axis is positioned on the screen.
44-45		The horizontal axis is positioned on the screen.
46-47		A message in FORMAT statement 110 is written for systems programming use.
48-55		Loop 200 constructs a step function of the predicted proportion of successes for each x-dosage arranged in ascending order.

SUBROUTINE GRACDF (Continued)

STATEMENT NUMBER	FORTRAN STATEMENT NUMBER	COMMENT
56-62		Loop 201 places asterisks on the diagram to denote the observed proportion of successes for each dosage.
63-69		Loop 16 constructs the markings on the vertical scale and writes the proportion of successes in tenths below this axis.
70-82		Loop 17 constructs the markings on the horizontal axis and writes the number of the batch for each x-dosage below this axis.
83-90		The minimum and maximum dosage-values are positioned on the lower portion of the screen.
91		The size routine is employed to determine whether or not the orders of display variable DVD will fit into the space that is available in the Buffer Table.
92		If the orders of display variable DVD will not fit into the available space in the BT, proceed to FORTRAN Statement 50.
93		Display variable DVD is plotted on the screen.
94	75	Procedure is detained until the user wishes to continue.
95		If an interrupt has been given by the user from any device except the programmed function key-

SUBROUTINE GRACDF (Continued)

STATEMENT NUMBER	FORTRAN STATEMENT NUMBER	COMMENT
		board (K is not equal to 1), proceed to detain statement 75.
96		If programmed function key (PFK) 31 has been depressed (NDET(4) equals 31), indicating the user wishes to terminate the program, proceed to FORTRAN statement 750 (exit).
97		If PFK 30 has been depressed (NDET(4) equals 30), indicating the user wishes to restart the program, proceed to FORTRAN statement 67.
98		If PFK 1 has been depressed (NDET(4) equals 1), indicating the user wishes to proceed in the program, proceed to FORTRAN statement 74.
99		Return to detain statement 75.
100	750	KEY is set equal to 50.
101		Proceed (branch around Restart).
102	67	Restart.
103-104	50	II, the size of space neccessary to display the display variable DVD is written for the programmer's use.
105	74	The screen is cleared of all existing displays.
106		Display variable DVD is re-initialized.

SUBROUTINE OUTPUA

Purpose: The purpose of the OUTPUA subroutine is to display detailed numerical results including the size of each batch, the observed proportion of successes, the predicted proportion of success under the quantal model, the predicted y-scores (Probits, etc.), the observed y-scores, the "working" quantities, the weights, the observed x-dosages, and the error of prediction of the y-scores for each class. A maximum of ten classes is presented on the screen at a time -- five classes on the upper portion of the screen and five more classes on the bottom half of the screen. This section has been divided into "pages" so that the user may return to the first page of the output at any time by depressing programmed function key 4.

Input:

P - a single-dimensioned field, in floating point, containing the observed proportion of successes of each batch

PCAP - a single-dimensioned field, in floating point, containing the predicted proportion of successes in each batch

YPR - a single-dimensioned field, in floating point, containing the predicted y-scores for each batch

Y - a single-dimensioned field, in floating point, containing the observed y-scores for each batch

YWORK - a single-dimensioned field, in floating point, containing the "working" quantities of each batch

W - a single-dimensioned field, in floating point, containing the weights in each batch

ER - a single-dimensioned field, in floating point, containing the error of prediction of the y-scores in each batch

The above input variables are stored on direct access disk 28.

A - a single-dimensioned field, containing the names of the five transformations available to the user- defined in the DATA statement.

Procedure:

STATEMENT NUMBER	FORTRAN STATEMENT NUMBER	COMMENT
5		Display variable DVA is established.
6-12		P, PCAP, YPR, Y, YWORK, W, ER are read from the direct access disk 28.
13-14		IXII and IXI are calculated, to be employed later in data field A.
15	102	NL, indicating the number of classes to be displayed, is set equal to NCL.
16		JKEY is set equal to zero. JKEY = 1 indicates a completed page of the calibration chart.
17		JEY is set equal to zero. JEY = 1 indicates that this is the last page of the output.
18		I, indicating the page number, is initialized equal to one.
19		K is set equal to zero. K is an even number indicating the top portion of the screen.
20		NI, indicating the increment of the classes to be displayed, is set equal to one.

SUBROUTINE OUTPUA (Continued)

STATEMENT NUMBER	FORTRAN STATEMENT NUMBER	COMMENT
21	94	If NL is less than 10, JEY is set equal to 1. NL is the number of classes left to be displayed.
22	1	If NL, the number of classes left to be displayed, is greater than 5, proceed to FORTRAN statement 2.
23		JKEY = 1 indicates a completed page of output.
24-26		NN controls the amount displayed in each section (two sections per page).
27-29	3	Decision whether one or two sections are needed on the page.
30-59	6	Output via display variable DVA.
61-63	11	Sequencing of pages to be displayed.
64		If JKEY is equal to 1, indicating completed page of calibration chart, proceed to FORTRAN statement 74; otherwise, proceed to FORTRAN statement 94.
65	9	Display variable DVA is plotted on the screen.
66	100	Procedure is detained to permit the user to change pages.
67		Display variable DVA is re-initialized.
68		If an interrupt has been given by the user from any device except the programmed function keyboard (J is not equal to 1), return to detain statement 100.

SUBROUTINE OUTPUA (Continued)

STATEMENT NUMBER	FORTRAN STATEMENT NUMBER	COMMENT
69		If programmed function key (PFK) 31 has been depressed (NDET(4) equals 31), indicating the user wishes to terminate the program, proceed to FORTRAN statement 750 (exit).
70		If PFK 30 has been depressed (NDET(4) equals 30), indicating the user wishes to restart the program, proceed to FORTRAN statement 67.
71		If PFK 1 has been depressed (NDET(4) equals 1), indicating the user wishes to continue to the next page of the output, proceed to FORTRAN statement 101.
72		If PFK 4 has been depressed (NDET(4) equals 4), indicating the user wishes to return to page 1 of the output, proceed to FORTRAN statement 102.
74	101	The scope is cleared of all existing displays.
76	4	I2 - the next page - is equal to I + 1.
77		The scope is cleared of all existing displays.
78-80		The title message, in FORMAT statement 5, is plotted on the screen.
81		If JEY equals 1, indicating the last page of the output, proceed to FORTRAN statement 12.
82-84		The sub-title message, in FORMAT statement 60 is written onto display variable DVA.

SUBROUTINE OUTPUA (Continued)

STATEMENT NUMBER	FORTRAN STATEMENT NUMBER	COMMENT
85		I, the page number, is incremented by one.
87-89	12	The sub-title message, in FORMAT statement 65 is written onto display variable DVA.
91	750	KEY is set equal to 50.
93	67	KEY is set equal to 1.
94	74	The scope is cleared of all existing displays.
95		Display variable DVA is re-initialized.

SUBROUTINE OUTPUT

Purpose: The purpose of the OUTPUT subroutine is two-fold:

(1) display an analysis of variance table and

(2) display the graph of the straight line $Y = \alpha + \beta x$

where α and β have been estimated in the analysis.

Input:

ALF - a floating point number obtained from the analysis, which estimates α

BETA - a floating point number obtained from the analysis, which estimates β

SSR - a floating point number obtained from the analysis which is the sum-of-squares for regression

SSE - a floating point number obtained from the analysis which is the sum-of-squares for error

Y - a single-dimensioned field, in floating point containing the y-scores for each class

The above input variables are obtained from a direct access disk.

A - a single-dimensioned field, containing the names of the five transformations available to the user - defined in the DATA statement

Procedure:

STATEMENT NUMBER	FORTRAN STATEMENT NUMBER	COMMENT
5		Display variable DVB is established.
6		The value of KEY is tested to determine if the user is returning to the QUANTAL program after calling the CALCG program (KEY = 25). If the value of KEY is 25, even these values

SUBROUTINE OUTPUT (Continued)

STATEMENT NUMBER	FORTRAN STATEMENT NUMBER	COMMENT
		which were in COMMON need to be re-read:
7-10		The values of X, N, NR, NCL and ITRAN are read from direct access disk 28.
12	1000	The field Y is read from direct access disk 28.
13-14	2000-2001	Variables ALF, BETA, SSR, and SSE are read into core.
15-16		IXI and IXII are calculated to obtain the correct subscript for field A.
17		JR, the degrees of freedom for regression, is set equal to 1.
18		JE, the degrees of freedom for error, is set equal to NCL-2.
19		JTOT, the total degrees of freedom is calculated.
20		On return from dosage-transformation routine, the analysis of variance display is skipped.
21		VSE, the mean squares for error is calculated.
22		STOT, the sum-of-squares total is calculated.
23		The F-statistic is calculated.
24-30		FORMAT statements 71 and 72, displaying the Analysis of Variance table, are plotted on the screen.
31	73	Procedure is detained until user has read the analysis of Variance table.

SUBROUTINE OUTPUT (Continued)

STATEMENT NUMBER	FORTRAN STATEMENT NUMBER	COMMENT
32		If an interrupt has been given by the user from any other device than the programmed function keyboard (I is not equal to 1), return to detain statement 73.
33		If programmed function key (PFK) 31 has been depressed by the user (NDET(4) equals 31), indicating the user wishes to terminate the program, proceed to FORTRAN statement 750 (exit).
34		If PFK 30 has been depressed (NDET(4) equals 30), indicating the user wishes to restart the program, proceed to FORTRAN 67.
35		If PFK 11 has been depressed (NDET(4) equals 11), indicating the user wishes to leave the QUANTAL program, proceed to FORTRAN statement 80.
36		If any PFK other than key 1 or those previously mentioned, has been depressed (NDET(4) is not equal to 1), return to detain statement 73.
37	491	The scope is cleared of all existing displays.
38		Display variable DVB is re-initialized.
39-41		FORMAT statement 400, a title message, is written onto display variable DVB.

SUBROUTINE OUTPUT (Continued)

STATEMENT NUMBER	FORTRAN STATEMENT NUMBER	COMMENT
42-44	2002	The message in FORMAT statement 2004 is written onto display variable DVB.
45-52		Loop 4 - The maximum of the x-values and the maximum of the y-values are determined.
53-60		In loop 8, the minimum of the x-values and the minimum of the y-values are determined.
61-71		XLGTH, XINC, YLGTH, YINC, Y2MIN, Y2MAX and X2MAX are calculated to determine the coordinates of the screen.
72		The coordinates of the lower left corner of the screen (XMIN, Y2MIN) and the upper right corner (X2MAX, Y2MAX) are established by the UCORD statement.
73-74		The vertical axis is positioned on the screen.
75-76		The horizontal axis is positioned on the screen.
77	13	Display variable DVB is plotted on the screen.
78-79		Diagnostic.
80	14	The length of the x-axis is divided into 200 equal lengths (XLGTH).
81		If KEY equals 6, indicating the x-dosages have been altered by one of the functions, proceed to FORTRAN statement 270 and skip loop 15.

SUBROUTINE OUTPUT (Continued)

STATEMENT NUMBER	FORTRAN STATEMENT NUMBER	COMMENT
82-88		In loop 15, the points on the straight line $Y = \alpha + \beta x$ are calculated and positioned onto display variable DVB.
89-92	270	Loop 16 positions an asterisk onto display variable DVB at each observed y-value.
93-94		XX and XNUM are calculated to determine ten equally spaced x-values on the horizontal axis.
97-98	36	XX and XNUM are calculated to determine ten equally spaced y-values on the vertical axis.
99-136	26	Loop 24 positions the markings on the horizontal scale when JEY equals 1 and the markings on the vertical scale when JEY equals 2 at ten equally spaced values. The x- and y-values are aligned by these markings by the five different FORMAT statements.
137		If JEY is not equal to 1, indicating both axes have been marked, proceed to FORTRAN statement 22.
138		JEY is set equal to 2, to initiate y-axis.
140	22	The size routine is employed to determine whether or not the orders of display variable DVB will fit into the space that is available in the BT.

SUBROUTINE OUTPUT (Continued)

STATEMENT NUMBER	FORTRAN STATEMENT NUMBER	COMMENT
141		If the orders of display variable DVB will not fit into the available space in the BT, proceed to FORTRAN statement 50.
142		Display variable DVB is plotted on the screen.
143-144	50	Diagnostic.
145	75	Procedure is detained until the user has read the display.
146		If an interrupt has been given by the user from any other device except the programmed function keyboard (K is not equal to 1), return to detain statement 75.
147		If PFK 31 has been depressed (NDET(4) equals 31), indicating the user wishes to terminate the program, proceed to FORTRAN statement 750 (exit).
148		If PFK 30 has been depressed (NDET(4) equals 30), indicating the user wishes to restart the program, proceed to FORTRAN statement 67.
149		If PFK 1 has been depressed (NDET(4) equals 1), indicating the user wishes to proceed in the program, exit from the subroutine.
151-154	80	X, N, NR, NCL and ITRAN are written onto direct access disk 28.

SUBROUTINE OUTPUT (Continued)

STATEMENT NUMBER	FORTRAN STATEMENT NUMBER	COMMENT
155		KEY = 50 to leave QUANTAL in order to do calculations.
157	750	Prepare for termination.
159	67	KEY is set equal to 1 (user wants to restart).
160	99	The screen is blanked of all existing displays.
161		Display variable DVB is re-initialized.

SUBROUTINE CHNGFN

Purpose: The purpose of the CHNGFN subroutine is to make various functions available for possible transformations of the x-dosages.

Procedure:

STATEMENT NUMBER	FORTRAN STATEMENT NUMBER	COMMENT
3		The display variables DVA, DVB, and DVE are established.
4		Display variable DVB is re-initialized.
5		The scope is cleared of all existing displays.
6-9		FORMAT statement 5 is plotted on the screen.
10	1	Procedure is detained until the user decides whether or not he wishes to change the x-dosages.
11		If an interrupt has been given by the user from any device except the programmed function keyboard (J is not equal to 1), return to detain statement 1.
12		If programmed function key (PFK) 31 has been depressed (NDET(4) equals 31), indicating the user wishes to terminate the program, proceed to FORTRAN statement 750.
13		If PFK 30 has been depressed (NDET(4) equals 30), indicating the user wishes to restart the program, proceed to FORTRAN statement 99 (exit).

SUBROUTINE CHNGFN (Continued)

STATEMENT NUMBER	FORTRAN STATEMENT NUMBER	COMMENT
14		If PFK 1 has been depressed (NDET(4) equals 1), indicating the user wishes no alteration to be made on the x-dosages, proceed to FORTRAN statement 50.
15		If PFK 3 has been depressed (NDET(4) equals 3), indicating the user desires to alter the x-dosages by one of the functions, proceed to FORTRAN statement 10.
17-20	10	The statement to obtain which function the user wishes (FORMAT 2) is plotted on the screen.
21		A cursor is placed in the first unprotected position of FORMAT statement 2 so that the user can type in the function desired.
22	3	Procedure is detained until the user types in the appropriate function.
23		If PFK 31 has been depressed, (NDET(4) equals 31), indicating the user wishes to terminate the program, proceed to FORTRAN statement 750 (exit).
24		If PFK 30 has been depressed (NDET(4) equals 30), indicating the user wishes to restart the program, proceed to FORTRAN statement 99.

SUBROUTINE CHNGFN (Continued)

STATEMENT NUMBER	FORTRAN STATEMENT NUMBER	COMMENT
25		If an interrupt has been given by the user from any device except the alphameric typewriter (J is not equal to 2), return to detain statement 7.
26-28		XFUN, the real value for IFUN is read.
29		The pointers for dummy unit 4 are reset.
30		IFUN = XFUN.
31		If the number of the function desired to alter the x-dosages is less than zero or greater than 12, proceed to FORTRAN statement 20 to write an error message.
32		When IFUN has the value 8, 9, or 10 proceed to FORTRAN statement 1050 to ascertain that all x-dosages are non-negative real-numbers. If IFUN has any other value (less than 12), proceed to FORTRAN statement 53.
33	53	KEY is set equal to 6, indicating subroutine FNS should be called.
35	20	Incorrectly answered FORMAT statement 2 is cleared from the screen.
36		Display variable DVA is re-initialized.
37-40		The error message in FORMAT 6 is plotted on the screen.

SUBROUTINE CHNGFN (Continued)

STATEMENT NUMBER	FORTRAN STATEMENT NUMBER	COMMENT
41		Display variable DVE is re-initialized.
43-45	1050	Loop 55 tests all x-values to ascertain they are non-negative real numbers. If any x-value is negative, proceed to write an error message.
47	52	Display variable DVA is re-initialized.
48		The current instance of DVA is removed from the Buffer Table.
49-52		The message in FORMAT statement 51 which states that the chosen function cannot be employed is plotted on the screen.
53		Display variable DVA is re-initialized.
55	750	KEY is set equal to 50.
57	99	KEY is set equal to 1.
59	50	KEY is set equal to 5.
60	100	The scope is cleared of all existing displays.
61		Display variables DVA, DVB, and DVE are re-initialized.

SUBROUTINE FNS

Purpose: The purpose of subroutine FNS is to transform the x-dosages by one of twelve mathematical functions made available to the user in CHNGFN.

Procedure: Loop 4 is executed for each x-dosage. The function employed is determined by the value of IFUN (obtained in subroutine CHNGFN) in the computed GO TO statement.

FIGURE 4.1

PROGRAM CONTROL CARDS

```
//QUANTAL JOB('ACCOUNT#',10,4),'PROGRAMMER',MSGLEVEL=1
//STEP1 EXEC FORTGC
//FORT.SYSLIN DD DSNAME=&&CHAIN(ROOT),SPACE=(TRK,(150,10,5)),
//                 UNIT=SYSDA,DISP=(NEW,PASS)
//FORT.SYSIN DD *
      (source deck for MAIN)
/*


//STEP2 EXEC FORTGC
//FORT.SYSLIN DD DSNAME=&&CHAIN(LINKA),DISP=(MOD,PASS),UNIT=SYSDA
//FORT.SYSIN DD *
      (source deck for DATA)
/*


//STEP3 EXEC FORTGC
//FORT.SYSLIN DD DSNAME=&&CHAIN(LINKD),DISP=(MOD,PASS),UNIT=SYSDA
//FORT.SYSIN DD *
      (source deck for INPUTA)
/*
```

FIGURE 4.1 (Continued)

```
//STEP4 EXEC FORTGC
//FORT.SYSLIN DD DSNAME=&&CHAIN(LINKE),DISP=(MOD,PASS),UNIT=SYSDA
//FORT.SYSIN DD *
       (source deck for INPUTB)
/*


//STEP5 EXEC FORTGC
//FORT.SYSLIN DD DSNAME=&&CHAIN(LINKI),DISP=(MOD,PASS),UNIT=SYSDA
//FORT.SYSIN DD *
       (source deck for CHNGTR)
/*


//STEP6 EXEC FORTGC
//FORT.SYSLIN DD DSNAME=&&CHAIN(LINKB),DISP=(MOD,PASS),UNIT=SYSDA
//FORT.SYSIN DD *
       (source deck for CALC)
/*


//STEP7 EXEC FORTGC
//FORT.SYSLIN DD DSNAME=&&CHAIN(LINKJ),DISP=(MOD,PASS),UNIT=SYSDA
//FORT.SYSIN DD *
       (source deck for GRACDF)
/*
```

FIGURE 4.1 (Continued)

```
//STEP8 EXEC FORTGC
//FORT.SYSLIN DD DSNAME=&&CHAIN(LINKF),DISP=(MOD,PASS),UNIT=SYSDA
//FORT.SYSIN DD *
      (source deck for OUTPUA)
/*


//STEP9 EXEC FORTGC
//FORT.SYSLIN DD DSNAME=&&CHAIN(LINKC),DISP=(MOD,PASS),UNIT=SYSDA
//FORT.SYSIN DD *
      (source deck for OUTPUT)
/*


//STEP10 EXEC FORTGC
//FORT.SYSLIN DD DSNAME=&&CHAIN(LINKG),DISP=(MOD,PASS),UNIT=SYSDA
//FORT.SYSIN DD *
      (source deck for CHNGFN)
/*


//STEP11 EXEC FORTGC
//FORT.SYSLIN DD DSNAME=&&CHAIN(LINKH),DISP=(MOD,PASS),UNIT=SYSDA
//FORT.SYSIN DD *
      (source deck for FNS)
/*
```

FIGURE 4.1 (Continued)

```
//STEP12 EXEC LKED,PARM=(LET,LIST,OVLY,XREF)
//LKED.SYSLMOD DD DSN=SYS1.GRAPHLIB(QUANTAL),DISP=SHR,
//              SPACE=(TRK,(0,0))
(GRAPHLIB(QUANTAL) is to receive the load module)
//LKED.SYSLIB DD DSN=SYS1.GRAFLIB,DISP=SHR
(GRAFLIB contains graphics routines)
//DD DSN=SYS1.FORTLIB,DISP=SHR
//DD DSN=SYS1.GRAPHLIB,DISP=SHR
//DD DSN=SYS1.UGALIB,DISP=SHR
//DD DSN=SYS1.LINKLIB,DISP=SHR
(utility routines at the University of Georgia system)
//LKED.MODULE DD DSN=&&CHAIN,DISP=OLD
(&&CHAIN contains the object modules, as prepared in the FORT step;
they are referred to under the DD name MODULE)
//LKED.SYSIN DD *
 INCLUDE MODULE(ROOT)    (contains MAIN program)
 INCLUDE SYSLIB(IBCOM#)
 INCLUDE SYSLIB(ARITH#)
 INCLUDE SYSLIB(FIOCS#)
 INCLUDE SYSLIB(ADCON#)
 INCLUDE SYSLIB(IHCUATBL)
 INCLUDE SYSLIB(IHCUOPT)
 INCLUDE SYSLIB(ERRMON)
 INCLUDE SYSLIB(IHCTRCH)
 INCLUDE SYSLIB(GAFERR)
```

FIGURE 4.1 (Continued)

```
OVERLAY ONE
INCLUDE SYSLIB(ARCOS)
INCLUDE SYSLIB(ATAN2)
INCLUDE SYSLIB(COS)
INCLUDE SYSLIB(DCOS)
INCLUDE SYSLIB(COTAN)
INCLUDE SYSLIB(DARCOS)
INCLUDE SYSLIB(DEXP)
INCLUDE SYSLIB(DLOG10)
INCLUDE SYSLIB(DSQRT)
INCLUDE SYSLIB(SQRT)
INCLUDE SYSLIB(ARSIN)
INCLUDE SYSLIB(ATAN)
INCLUDE SYSLIB(SIN)
INCLUDE SYSLIB(DSIN)
INCLUDE SYSLIB(TAN)
INCLUDE SYSLIB(DARSIN)
INCLUDE SYSLIB(DLOG)
OVERLAY TWOA
INCLUDE MODULE(LINKB)        (contains SUBROUTINE CALC)
OVERLAY TWOA
INCLUDE MODULE(LINKH)        (contains SUBROUTINE FNS)
OVERLAY ONE
INCLUDE SYSLIB(DISPLA)
INCLUDE SYSLIB(PLOT)
```

FIGURE 4.1 (Continued)

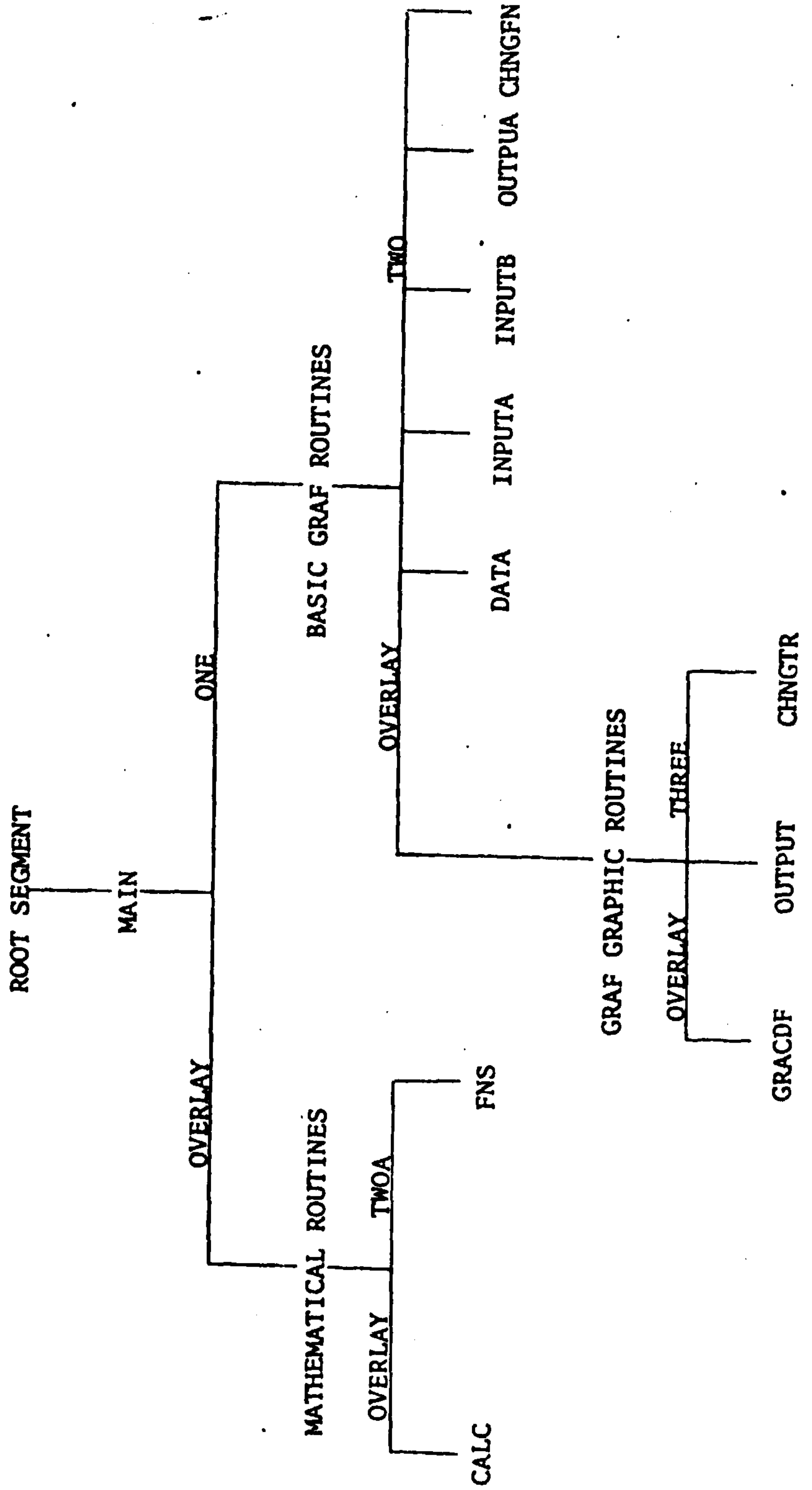
```
INCLUDE SYSLIB(DETAIN)
INCLUDE SYSLIB(BUFRS)
INCLUDE SYSLIB(CUR$$)
INCLUDE SYSLIB(LIGHTS)
INCLUDE SYSLIB(SCTDV)
INCLUDE SYSLIB(ERASE)
INCLUDE SYSLIB(RCUR$)
INCLUDE SYSLIB(RESET)
INCLUDE SYSLIB($$$$BT)
INCLUDE SYSLIB(CHAR)
INCLUDE SYSLIB(DETEKT)
INCLUDE SYSLIB(READSC)
INCLUDE SYSLIB(CLOSE)
INCLUDE SYSLIB($VOVER)
INCLUDE SYSLIB($$OVER)
INCLUDE SYSLIB(DUMMY$)
INCLUDE SYSLIB(UNPLOT)
INCLUDE SYSLIB(PLACE$)
INCLUDE SYSLIB(BLANK)
INCLUDE SYSLIB($ADD$)
INCLUDE SYSLIB($$INIT)
INCLUDE SYSLIB(SCNDVDK)
INCLUDE SYSLIB(WRFMT$)
OVERLAY TWO
INCLUDE MODULE(LINKA)      (contains SUBROUTINE DATA)
```

FIGURE 4.1 (Continued)

```
OVERLAY TWO
INCLUDE MODULE(LINKD)     (contains SUBROUTINE INPUTA)
OVERLAY TWO
INCLUDE MODULE(LINKE)     (contains SUBROUTINE INPUTB)
OVERLAY TWO
INCLUDE MODULE(LINKF)     (contains SUBROUTINE OUTPUA)
OVERLAY TWO
INCLUDE MODULE(LINKG)     (contains SUBROUTINE CHNGFN)
OVERLAY TWO
INCLUDE SYSLIB(CORDCALL)
INCLUDE SYSLIB(LINE$$)
INCLUDE SYSLIB(SIZE)
INCLUDE SYSLIB($CORD$)
INCLUDE SYSLIB(POINT$)
OVERLAY THREE
INCLUDE MODULE(LINKC)     (contains SUBROUTINE OUTPUT)
OVERLAY THREE
INCLUDE MODULE(LINKI)     (contains SUBROUTINE CHNGTR)
OVERLAY THREE
INCLUDE MODULE(LINKJ)     (contains SUBROUTINE GRACDF)
//
```

FIGURE 4.2

DIAGRAM OF OVERLAY STRUCTURE



REFERENCES

[1] Bargmann, R. E., Association in a Class of Growth Functions, University of Illinois, 1966.

[2] Dixon, W. J., "Use of Displays with Packaged Statistical Programs," Proceedings, AFIPS 1967 Fall Joint Computer Conference, MDI Publications, Wayne, Pa., pp. 481-484.

[3] Finney, D. J., Probit Analysis, Cambridge: Cambridge University Press (2nd edition), 1952.

[4] ______, Statistical Method in Biological Assay, New York: Hafner, 1952.

[5] Fisher, R. A., and Yates, F., Statistical Tables for Biological, Agricultural, and Medical Research, (3rd edition), New York: Hafner, 1948.

[6] Hilsenrath, J.; Guy C. Ziegler; C. G. Messiwa; P. J. Walsh; and R. J. Herbold. OMNITAB, a Computer Program for Statistical and Numerical Analysis. National Bureau of Standards, Handbook 101, 1966.

[7] Hurwitz, A.; Joseph Yeaton; and Roberta Schaffer. Graphics Addition to FORTRAN, 360 D-03,8.010. Los Angeles Scientific Center: IBM Corporation, 1968.

[8] IBM, System/360 Scientific Subroutine Package (360A - CM - 03X) Version III, Programmers' Manual, (4th edition), 1968.

[9] Nelder, J. A., "General Statistical Program (GENSTAT Version IV)." User's Guide, Biometry Section, Waite Institute, Glen Osmond, South Australia, 1966.

[10] Penn, L. W., An On-Line Statistical Computer System for Lay Usage, University of Georgia, 1971.

APPENDIX A

FORTRAN IV LISTING

NOT REPRODUCIBLE

```
FORTRAN IV G LEVEL  19                MAIN               DATE = 71056        13/17/12            PAGE 0001

0001                COMMON NOET(5),X(100),N(100),NR(100),DV1,DV2,DV3,DV4,DV5,DV6,DV7,
                   1DVD,DVE,DVA,DVB,NCL,KEY,KK,IFUN,ITRAN,IREC
0002                DEFINE FILE 28(13,400,U,IREC)
0003              1 KEY = 0
0004                CALL INPUTA
0005                IF(KEY.EQ.50)GO TO 99
0006                IF(KEY.EQ.25)GO TO 9
0007              5 CALL DATA
0008                IF(KEY.EQ.50)GO TO 99
0009              2 IF(KEY.EQ.1)GO TO 1
0010                CALL INPUTB
0011                IF(KEY.EQ.50)GO TO 99
0012                IF(KEY.EQ.1)GO TO 1
0013                IF(KEY.EQ.7)GO TO 3
0014                IF(KEY.EQ.10)GO TO 5
0015                GO TO 99
0016              3 CALL CHAGTR
0017                IF(KEY.EQ.50)GO TO 99
0018                IF(KEY.EQ.1)GO TO 1
0019              7 CALL CLOSES
0020                CALL CALC
0021              8 CALL GRACCF
0022                IF(KEY.EQ.50)GO TO 99
0023                IF(KEY.EQ.1)GO TO 1
0024                CALL OUTPUA
0025                IF(KEY.EQ.50)GO TO 99
0026                IF(KEY.EQ.1)GO TO 1
0027              9 CALL OUTPUT
0028                IF(KEY.EQ.50)GO TO 99
0029                IF(KEY.EQ.1)GO TO 1
0030              4 CALL CHACFN
0031                IF(KEY.EQ.50)GO TO 99
0032                GO TO (1,2,3,4,2,6),KEY
0033              6 CALL CLOSES
0034                CALL FNS
0035                GO TO 8
0036             99 STOP
0037                END
```

```
FORTRAN IV G LEVEL  19              INPUTA              DATE = 71056         13/17/31              PAGE 0001
0001                SUBROUTINE INPUTA
0002                COMMON NDET(5),X(100),N(100),NR(100),DV1,DV2,DV3,DV4,DV5,DV6,DV7,
                   1DV0,DVE,DVA,DVB,NCL,KEY,KK,IFUN,ITRAN,IRFC
0003                CALL DISPLA(DV1,DV2,DV3,DV4,DVE)
0004                MI= LIGHTS(1,1,2,3,4,6,7,8,9,10,30,31,5,11)
0005             67 CONTINUE
0006                CALL BLANK
0007                CALL RCURS
0008                CALL RESET(DV1,DV2,DV3,DV4,DVE)
0009                WRITE(4,400)
0010                CALL WRFMTS(DV1)
0011            400 FORMAT('P1'/'PC'/'PO      IF YOU ARE ENTERING THE  "QUANTAL"  PROGR
                   1AM FOR THE FIRST TIME,'/'P DEPRESS, PROGRAM FUNCTION KEY 1.'/'PO
                   2   IF YOU ARE RETURNING FROM THE "CALCG" PROGRAM AND WISH TO PROCE
                   3ED'/'P TO THE POINT OF YOUR DEPARTURE, DEPRESS PROGRAM FUNCTION KE
                   4Y 2.'/'PO      THE PROGRAM FUNCTION KEYBOARD IS LOCATED TO THE LEFT
                   5 OF THE'/'P TYPEWRITER KEYBOARD.')
0012                IBUF = PLOT(DV1)
0013            402 I = DETAIN(NDET)
0014                IF(NDET(4).EQ.2)GO TO 401
0015                IF(NDET(4).EQ.1)GO TO 66
0016                GO TO 402
0017             66 CALL BLANK
0018                CALL RESET(DV1)
0019                WRITE(4,100)
0020                CALL WRFMTS(DV1)
0021            100 FORMAT('P1'/'PC  THIS SECTION OF THE QUANTAL PROGRAM IS DESIGNED T
                   1O ASSIST YOU'/'P IN CONSTRUCTING A MODEL OF THE DESIGN TO BE STUDI
                   2ED.  THIS MODEL WILL'/'P BE CONSTRUCTED BY YOUR ANSWERING QUESTION
                   3S AND FILLING IN BLANKS.'/'PO      GENERAL INSTRUCTIONS OF THE PROG
                   4RAM:'/'P 1.  AT ANY POINT IN THE PROGRAM YOU MAY RESTART BY DEPRES
                   5SING PROGRAM'/'P FUNCTION KEY 30 OR TERMINATE BY DEPRESSING PROGRA
                   6M FUNCTION KEY 31.')
0022                WRITE(4,1000)
0023                CALL WRFMTS(DV1)
0024           1000 FORMAT(             'P 2.  YOUR ANSWERS SHOULD BE TYPED FROM THE TYP
                   7EWRITER KEYBOARD DIRECTLY'/'P IN FRONT OF YOU (UNLESS YOU ARE ASKE
                   8D DIFFERENTLY).'/'P 3.  ALL NUMBERS ARE TO BE TYPED AS REAL NUMBER
                   9S--DECIMAL POINT MUST '/'P FOLLOW THE NUMBER.'/'P 4.  AFTER EACH S
                   1TATEMENT, YOU SHOULD CAUSE AN "END" STATEMENT BY FIRST'/'P DEPRESS
                   2ING THE "ALTN CODING" KEY AND, WHILE IT IS HELD DOWN, DEPRESS THE'
                   3/'P "5" KEY.'/'P 5.  IF THERE IS MORE THAN ONE BLANK TO BE FILLED
                   4IN A LINE,'/'P DEPRESS THE "JUMP" KEY LOCATED ON THE LEFT SIDE OF
                   5THE'/'P TYPEWRITER KEYBOARD TO PROCEED FROM BLANK TO BLANK.'/'PO
                   6   WHEN YOU ARE READY TO PROCEED, DEPRESS PROGRAM FUNCTION KEY 1')
0025                IBUF = PLOT(DV1)
0026            201 I = DETAIN(NDET)
0027                IF(NDET(4).EQ.31)GO TO 750
0028                IF(NDET(4).EQ.30)GO TO 67
0029                IF(I.NE.1)GO TO 201
0030                CALL BLANK
0031             25 WRITE(4,101)
0032                CALL WRFMTS(DV2)
0033            101 FORMAT('P1        NOW, HOW MANY CLASSES (BATCHES) ARE THERE IN YOUR D
                   1ESIGN? '/'U.',5X/'P NOTE:  YOUR NUMBER MUST BE GREATER THAN 0 AND
                   2LESS THAN 100.')
```

```
0034            IBUF = PLOT(DV2)
0035            CALL CURSS(DV2,64)
0036        200 I = DETAIN(NDET)
0037            IF(I.EQ.1 .AND. NDET(4).EQ.31)GO TO 750
0038            IF(I.EQ.1 .AND. NDET(4).EQ.30)GO TO 67
0039         69 IF(I.NE.2)GO TO 200
0040            CALL SC'DV(DV2)
0041            CALL DVTCH(DV2)
0042            READ(4,300)XNUM
0043        300 FORMAT(/G5.0)
0044            CALL BUFRS
0045            K = UNPLOT(DV2)
0046            CALL RCURS
0047            IBUF = PLOT(DV2)
0048            NCL = XNUM
0049            IF(NCL.LE.0.OR.NCL.GT.100)GO TO 30
0050         50 K = ERASE(DVE)
0051            GO TO 21
0052         30 WRITE(4,114)
0053            CALL WRFPTS(DVE)
0054        114 FORMAT('P THE NUMBER OF CLASSES (BATCHES) MUST BE GREATER THAN 0 A
               1ND LESS THAN 100.')
0055            IBUF = PLOT(DVE)
0056            K = UNPLOT(DV2)
0057            CALL RESET(DV2)
0058            CALL RESET(DVE)
0059            GO TO 25
0060        401 KEY = 25
0061            CALL BLANK
0062            GO TO 21
0063        750 KEY = 50
0064         21 CALL RESET(DV1,DV2,DV3,DV4,DVE)
0065            RETURN
0066            END
```

```
FORTRAN IV G LEVEL  19              DATA              DATE = 71056        13/17/21              PAGE 0001

0001                SUBROUTINE DATA
0002                COMMON NDET(5),X(100),N(100),NR(100),DV1,DV2,DV3,DV4,DV5,DV6,DV7,
                   1DVC,DVE,CVA,CVE,NCL,KEY,KK,IFUN,ITRAN,IREC
0003                CALL DISPLA(CV3,CV4,DVE)
0004                CALL BLANK
0005                IF(KEY.EQ.10)GO TO 75
0006                NL = NCL
0007                JEY = C
0008                NI = 1
0009              4 WRITE(4,102) NCL
0010                CALL WRFMTS(CV3)
0011            102 FORMAT('P1       NEXT COMPLETE THE FOLLOWING STATEMENT FOR EACH OF T
                   1HE',I5/'P MATCHES.  AN EXAMPLE IS, IN BATCH 12, WHERE THE CONCOMIT
                   2ANT VARIABLE'/'P (DOSAGE)      6.4       WAS GIVEN TO 36. SUBJECTS,
                   3THERE WERE 3C. SUBJECTS'/'P THAT RESPONDED (SUCCESSES).'/'PO      T
                   4HE NUMBER OF SUBJECTS IN EACH BATCH MUST BE GREATER THAN 2.')
0012                IBUF = PLOT(CV3)
0013                IF(NL.GT. 8)GO TO 2
0014                JEY = 1
0015                NN = NCL
0016                GO TO 3
0017              2 NN = NI + 7
0018              3 DO 500 KK=NI,NN
0019             24 KEY = C
0020                WRITE(4,103)KK
0021                CALL WRFMTS(CV4)
0022            103 FORMAT('PO     IN BATCH ',I5,' WHERE THE CONCOMITANT VARIABLE (DOSA
                   1GE) '/'U.',14X/'P WAS GIVEN TO '/'U.',5X/'P. SUBJECTS, THERE WERE
                   2 '/'U.',5X/'P. SUBJECTS THAT RESPONDED'/'P (SUCCESSES).')
0023                IBUF = PLOT(DV4)
0024                CALL CURSS(DV4,60)
0025            201 I = DETA(NCEI)
0026                IF(I.EQ.1 .AND. NDET(4).EQ.31)GO TO 750
0027                IF(I.EQ.1 .AND. NDET(4).EQ.30)GO TO 67
0028                IF(I.NE.2)GO TO 201
0029                CALL SCTEV(DV4)
0030                CALL DVTOM(DV4)
0031                READ(4,301)X(KK),XX,XY
0032            301 FORMAT(/G14.7//G5.0//G5.0)
0033                CALL BUFRS
0034                N(KK)=XX
0035                NR(KK) = XY
0036             19 IF(KEY.EQ.10)GO TO 20
0037             22 IF(N(KK).LE.2)GO TO 11
0038             12 IF(NR(KK))13,15,15
0039             15 IF(NR(KK).GT.N(KK))GO TO 45
0040                GO TO 501
0041             20 WRITE(4,21)
0042                CALL WRFMTS(DVE)
0043             21 FORMAT('P1'/'PC')
0044                GO TO 22
0045             11 WRITE(4,117)
0046                CALL WRFMTS(DVE)
0047            117 FORMAT('P THE NUMBER OF SUBJECTS MUST BE GREATER THAN 2')
0048                IBUF = PLOT(DVE)
0049                K = UNPLOT(DV4)
```

```
      CALL RESET(CVE,DV4)
      GO TO 24
   13 WRITE(4,118)
      CALL WRFPTS(CVE)
  118 FORMAT('P THE NUMBER OF SUBJECTS THAT RESPONDED (SUCCESSES) MUST B
     1E GREATER THAN 0')
      IBUF = PLOT(CVE)
      K = UNPLOT(DV4)
      CALL RESET(CVE,DV4)
      GO TO 24
   45 WRITE(4,120)
      CALL WRFPTS(CVE)
  120 FORMAT('P THE NUMBER OF SUBJECTS THAT RESPONDED (SUCCESSES) MUST N
     1OT EXCEED THE'/'P NUMBER OF SUBJECTS')
      IBUF = PLOT(CVE)
      K = UNPLOT(DV4)
      CALL RCURS
      CALL RESET(CVE,DV4)
      GO TO 24
  501 K = UNPLOT(DV4)
      CALL RCURS
      IBUF = PLOT(CV4)
      CALL RESET(DV4)
  500 K = ERASE(DVE)
      CALL BLANK
      NI = NI + 8
      NL = NL - 8
      IF(JEY)4,4,99
   75 NN = KK
      NI = NN
      JEY = 1
      WRITE(4,192)
      CALL WRFPTS(CV3)
  192 FORMAT('P1')
      IBUF = PLOT(CV3)
      GO TO 3
  750 KEY = 50
      GO TO 99
   67 KEY = 1
   99 CALL RESET(DV3,CV4,DVE)
   94 RETURN
      END
```

```
0001           SUBROUTINE INPLTB
0002           COMMON NDET(5),X(100),N(100),NR(100),DV1,DV2,DV3,DV4,DV5,DV6,DV7,
              1DVD,DVE,DVA,DVB,NCL,KEY,KK,IFUN,ITRAN,IREC
0003           CALL DISPLA(DV5,DV6,DV7,DVD,DVA,DVE)
0004       502 CALL BLANK
0005           WRITE(4,104)
0006           CALL WRFMTS(DV5)
0007       104 FORMAT('P1'/'PC'/'PO      YOUR DATA NOW IS:'/
              1                                  'P BATCH NUMBER      DOSAGE      N
              1UMBER OF SUBJECTS      NUMBER OF SUCCESSES')
0008           K = 0
0009           DO 21 JJ=1,NCL
0010           WRITE(4,105)JJ,X(JJ),N(JJ),NR(JJ)
0011           CALL WRFMTS(DV5)
0012       105 FORMAT('P       ',I5,/'P.      ',F10.5,/'P.        ',I5,/'P.
              1           ',I5)
0013           IBUF = PLOT(DV5)
0014           CALL RESET(DV5,DV7)
0015           K = K + 1
0016           IF(K.NE.25)GO TO 21
0017           CALL BLANK
0018           K = 0
0019           WRITE(4,104)
0020           CALL WRFMTS(DV5)
0021        21 CONTINUE
0022           WRITE(4,106)
0023           CALL WRFMTS(DV7)
0024       106 FORMAT('PC      IF THERE IS ANY CORRECTION TO YOUR DATA, DEPRESS PR
              1OGRAM FUNCTION'/'P KEY 2.  IF YOU WISH TO ADD ANOTHER CLASS (BATCH
              2) DEPRESS PROGRAM'/'P FUNCTION KEY 5.  IF YOU WISH TO CONTINUE, DE
              3PRESS PROGRAM FUNCTION'/'P KEY 1.')
0025           IBUF = PLOT(DV7)
0026       202 J = DETAIN(NCET)
0027           IF(J.NE.1)GO TO 202
0028         7 K = ERASE(DVE)
0029           IF(NDET(4).EQ.31)GO TO 750
0030         8 IF(NDET(4).EQ.30)GO TO 67
0031           IF(NDET(4).EQ.1)GO TO 76
0032           IF(NDET(4).EQ.2)GO TO 68
0033           IF(NDET(4).EQ.5)GO TO 300
0034           GO TO 202
0035       300 NCL = NCL + 1
0036           CALL BLANK
0037           KK = NCL
0038           KEY = 10
0039           GO TO 74
0040        68 CALL BLANK
0041           CALL RESET(DVA)
0042           WRITE(4,203)
0043           CALL WRFMTS(DVA)
0044       203 FORMAT('P1      WHICH CLASS (BATCH) DO YOU WISH TO CORRECT? '/'U.',
              15X)
0045           IBUF = PLOT(DVA)
0046           CALL CURSS(DVA,50)
0047       206 I = DETAIN(NCET)
0048           IF(I.EQ.1 .AND. NDET(4).EQ.31)GO TO 750
```

```
      IF(I.EQ.1 .AND. NDET(4).EQ.30)GO TO 67
      IF(I.NE.2)GO TO 206
      CALL SCTCV(DVA)
      CALL DVTDM(DVA)
      READ(4,207)XK
  207 FORMAT(/G5.0)
      CALL BUFRS
      KK = XK
      IF(KK.LT.0 .OR. KK.GT.NCL)GO TO 209
      K = UNPLCT(DVA)
      CALL KCURS
      IBUF = PLCT(DVA)
      KEY = 10
      GO TO 74
  209 WRITE(4,211)NCL
      CALL WRFMTS(DVE)
  211 FORMAT('P THE CORRECTED CLASS (BATCH) NUMBER MUST BE GREATER THAN
     10 AND LESS THAN ',I5)
      IBUF = PLCT(DVE)
      K = UNPLCT(DVA)
      CALL RESET(DVE)
      GO TO 68
  750 KEY = 50
      GO TO 74
   76 KEY = 7
      GO TO 74
   67 KEY = 1
   74 CALL BLANK
      CALL RESET(DV1,DV2,DV3,DV4,DV5,DV6,DV7,DVD,DVE)
      RETURN
      END
```

```
FORTRAN IV G LEVEL  19                CHNGTR              DATE = 71056          13/18/31             PAGE 0001

            SUBROUTINE CHNGTR
            COMMON NDET(5),X(100),N(100),NR(100),DV1,DV2,DV3,DV4,DV5,DV6,DV7,
           1DVD,DVE,DVA,DVB,NCL,KEY,KK,IFUN,ITRAN,IREC
            CALL DISPLA(DV1)
            WRITE(4,1)
            CALL WRFMTS(DV1)
          1 FORMAT('P1'/'PC      NOW CHOOSE THE TRANSFORMATION BELOW WHICH YOU
           1DESIRE AND DEPRESS THE'/'P CORRESPONDING PROGRAM FUNCTION KEY.'/'P
           20'/'PO',10X,'PROBIT TRANSFORMATION    -  PROGRAM FUNCTION KEY 6'/'P
           3 ',10X,'LOGIT TRANSFORMATION     -  PROGRAM FUNCTION KEY 7'/'P ',10
           4X,'LOG-LOG TRANSFORMATION  -  PROGRAM FUNCTION KEY 8'/'P ',10X,
           5'ARC SIN TRANSFORMATION  -  PROGRAM FUNCTION KEY 9'/'P ',10X,'WEIB
           6ULL TRANSFORMATION  -  PROGRAM FUNCTION KEY 10')
            IBUF = PLOT(DV1)
          2 J = DETAIN(NDET)
            IF(J.NE.1)GO TO 2
            IF(NDET(4).EQ.31)GO TO 750
            IF(NDET(4).EQ.30)GO TO 67
            IF(NDET(4).EQ.6)ITRAN=1
            IF(NDET(4).EQ.7)ITRAN=2
            IF(NDET(4).EQ.8)ITRAN=3
            IF(NDET(4).EQ.9)ITRAN=4
            IF(NDET(4).EQ.10)ITRAN=5
            IF(NDET(4).LT.6.OR.NDET(4).GT.10)GO TO 2
            GO TO 99
        750 KEY = 50
            GO TO 99
         67 KEY = 1
         99 CALL RESET(DV1)
            CALL BLANK
            RETURN
            END
```

```
0001            SUBROUTINE CALC
0002            DOUBLE PRECISION DPI,DYI,DFX,DFP,DFZ,YORMX,YORMP,YORPZ,DFX3,DFP3,
               1DFZ3,DFX4,DFP4,DFZ4,DFX2,DFP2,DFZ2
0003            DIMENSION U(100),Z(100),P(100),PCAP(100),YPR(100),Y(100),YWORK(100
               1),W(100),ER(100)
0004            COMMON NDET(5),X(100),N(100),NR(100),DV1,DV2,DV3,DV4,DV5,DV6,DV7,
               1DV8,DVE,DVA,CVE,NCL,KEY,KK,IFUN,ITRAN,IREC
0005            KMAX = 0
0006            K = 0
0007            ALFO = 0.
0008            BETO = 0.
0009       101  IF(K)99,3,21
0010         3  DO 58 I=1,NCL
0011        58  P(I) = FLOAT(NR(I))/FLOAT(N(I))
0012            DO 5 I=1,NCL
0013            IF(P(I)) 6,6,7
0014         6  P(I) = 0.5/FLOAT(N(I))
0015            GO TO 5
0016         7  IF(P(I) - 0.99999) 5,8,8
0017         8  P(I) = 1.0 - 0.5/FLOAT(N(I))
0018            GO TO 5
0019         5  CONTINUE
0020        77  DO 11 I=1,NCL
0021            DPI = P(I)
0022            GO TO (50,51,52,53,76),ITRAN
0023         50 Y(I) = YORMP(DPI)
0024            GO TO 11
0025         51 Y(I) = DFP(DPI)
0026            GO TO 11
0027         52 Y(I) = DFP3(DPI)
0028            GO TO 11
0029         53 Y(I) = DFP4(DPI)
0030            GO TO 11
0031         76 Y(I) = DFP2(DPI)
0032         11 CONTINUE
0033            IF(K) 99,10,21
0034        10  DO 12 I=1,NCL
0035            YWORK(I) = Y(I)
0036        12  W(I) = N(I)
0037            GO TO 38
0038        21  DO 23 I=1,NCL
0039            DYI = YPR(I)
0040            GO TO (54,55,56,57,78),ITRAN
0041         54 Z(I) = YORMZ(DYI)
0042            GO TO 59
0043         55 Z(I) = DFZ(DYI)
0044            GO TO 59
0045         56 Z(I) = DFZ3(DYI)
0046            GO TO 59
0047         57 Z(I) = DFZ4(DYI)
0048            GO TO 59
0049         78 Z(I) = DFZ2(DYI)
0050         59 CONTINUE
0051            W(I) = FLOAT(N(I))*Z(I)**2/(PCAP(I)*(1.-PCAP(I)))
0052        23  YWORK(I) = YPR(I) + (P(I)-PCAP(I))/Z(I)
0053        38  IF(KMAX) 99,41,42
```

```
FORTRAN IV G LEVEL  19                CALC              DATE = 71056         13/17/47              PAGE 0002

CC54           42   IF(K - KMAX) 41,99,99
CC55           41   IF(K) 99,84,84
CC56           84   CALL WREG(X,YWORK,Y,W,NCL,YPR,ER,SSX,SSY,SPXY,SSR,SSE,VSE,R,ALF,BE
                   1TA,SW,SUMX,SUMY,XBAR,YBAR)
CC57                K = K + 1
CC58                IF(ABS((ALF-ALFO)/ALF).GT.1.E-4 .OR. ABS((BETA-BETO)/BETA).GT.1.E-
                   14)  GO TO 60
CC59                IF(KMAX)1006,1006,60
CC60           60   ALFO = ALF
CC61                BETO = BETA
CC62                IF(K- 5) 141,99,99
CC63          141   DO 65 I=1,NCL
CC64                DYI = YPR(I)
CC65                GO TO (71,72,73,74,75),ITRAN
CC66           71   PCAP(I) = YORMX(DYI)
CC67                GO TO 65
CC68           72   PCAP(I) = DFX(DYI)
CC69                GO TO 65
CC70           73   PCAP(I) = DFX3(DYI)
CC71                GO TO 65
CC72           74   PCAP(I) = DFX4(DYI)
CC73                GO TO 65
CC74           75   PCAP(I) = DFX2(DYI)
CC75           65   CONTINUE
CC76                GO TO 101
CC77         1006   CONTINUE
CC78           99   WRITE(28'1)P
CC79                WRITE(28'2)PCAF
CC80                WRITE(28'3)YPR
CC81                WRITE(28'4)Y
CC82                WRITE(28'5)YWORK
CC83                WRITE(28'6)W
CC84                WRITE(28'7)ER
CC85                WRITE(28'8)SSR,SSE
[redacted]
CC98                WRITE(28'9)ALF,BETA
[redacted]
CC92         2001   RETURN
CC93                END
```

```
C     WEIGHTED REGRESSION SUBROUTINE. THE ARRAY Y IS TAKEN AS THE
C      WORKING ARRAY. IF IT IS DIFFERENT FROM THE DATA, THE ARRAY YDATA
C       SHOULD BE THE ORIGINAL DATA. IN SIMPLE WEIGHTED REGRESSION
C      BOTH Y AND YDATA SHOULD CONTAIN THE SAME LISTS
C-----------------------------------------------------------------------
      SUBROUTINE WREG(X,Y,YDATA,W,NCL,YCAL,ER,SSX,SSY,SPXY,SSR,SSE,VSE,R
     1,ALF,BETA,SW,SUMX,SUMY,XBAR,YBAR)
      DIMENSION X(1),Y(1),W(1),YCAL(1),ER(1),YDATA(1)
      SUMX = 0.
      SUMY = 0.
      SSX = 0.
      SSY = 0.
      SPXY = 0.
      SW = 0.
      DO 1 I=1,NCL
      SUMX = SUMX + W(I)*X(I)
      SUMY = SUMY + W(I)*Y(I)
    1 SW = SW + W(I)
      XBAR = SUMX/SW
   11 YBAR = SUMY/SW
   13 DO 2 I=1,NCL
      X(I) = X(I) - XBAR
    2 Y(I) = Y(I) - YBAR
      DO 3 I=1,NCL
      SSX = SSX + W(I)*X(I)**2
      SSY = SSY + W(I)*Y(I)**2
    3 SPXY = SPXY + W(I)*X(I)*Y(I)
   15 BETA = SPXY/SSX
      SSR = SPXY**2/SSX
   16 SSE = SSY - SSR
      VSE = SSE/FLOAT(NCL-2)
      DO 5 I=1,NCL
    5 YCAL(I) = YBAR + BETA*X(I)
      DO 4 I=1,NCL
      X(I) = X(I) + XBAR
    4 Y(I) = Y(I) + YBAR
      DO 6 I=1,NCL
    6 ER(I) = YDATA(I) - YCAL(I)
      R = SIGN(SQRT(SSR/SSY),BETA)
      ALF = YBAR - BETA*XBAR
      RETURN
      END
```

```
FORTRAN IV G LEVEL  19                YORMX              DATE = 71056          13/17/47              PAGE 0001

            DOUBLE PRECISION FUNCTION YORMX(DZ)
            IMPLICIT  REAL*8(D)
            DOUBLE PRECISION YORMX
            DPI = .3989422804014 33
            DX = DABS(DZ)
            IF(DX.GT.3.D0) GO TO 10
            DAL = 0.D0
            DBL = 1.D0
            DAH = DX
            DBH = 1.D0
            DAN = 0.D0
      5     DAN = DAN + 1.D0
            DAI = -(2.D0*DAN - 1.D0) *DX*DX
            DBI = 4.D0*DAN - 1.D0
            DAL = DBI*DAH + DAI*DAL
            DBL = DBI*DBH + DAI * DBL
            DAI = DX*DX - DAI
            DBI = 2.D0 + DBI
            DAH = DBI*DAL + DAI*DAH
            DBH = DBI*DBL + DAI*DBH
            DFA = DAL/DBL
            DFB = DAH/DBH
            IF(DFB.EQ.0.D0) GO TO 20
            IF(DABS((DFB-DFA)/DFA).LE.1.D-14) GO TO 20
            GO TO 5
     10     DAL = 0.D0
            DBL = 1.D0
            DAH = 1.D0
            DBH = DX
            DBI = DX
            DAN = 1.D0
            DFA = 1.D0/DX
     15     DAN = DAN + 1.D0
            DAI = DAN - 1.D0
            DAC = DBI*DAH + DAI*DAL
            DBC = DBI*DBH + DAI*DBL
            DFB = DAC/DBC
            DAL = DAH
            DBL = DBH
            DAH = DAC
            DBH = DBC
            IF(DFB.EQ.0.D0) GO TO 20
            IF(DABS((DFB-DFA)/DFB).LE.1.D-14) GO TO 20
            DFA = DFB
            GO TO 15
     20     YORMX = DPI*DFB*DEXP(-DX*DX/2.D0)
            IF(DX.LE.3.D0) YORMX = 0.5D0 - YORMX
            IF(DZ.GT.0.D0) YORMX = 1.D0 - YORMX
            RETURN
            END
```

```
FORTRAN IV G LEVEL  19                MAIN              DATE = 71056          13/17/47              PAGE 0001

                C     DOUBLE PRECISION NORMAL DISTRIBUTION--ARGUMENT P, RESULT X.
                C     YORMP REQUIRES FUNCTION SUBROUTINES YORMZ AND YORMX
 0001                 DOUBLE PRECISION FUNCTION YORMP(DP)
 0002                 IMPLICIT REAL*8(D,Y)
 0003                 DOUBLE PRECISION YORMZ
 0004                 Y=DP
 0005                 DOMEG = .5399995999999999999D38
 0006                 IF(Y)1,1,2
 0007               1 YORMP = -DOMEG
 0008                 GO TO 99
 0009               2 IF(Y-1.D0)3,4,4
 0010               4 YORMP = DOMEG
 0011                 GO TO 99
 0012               3 IF(Y-.5D0)5,6,7
 0013               6 YORMP = 0.D0
 0014                 GO TO 99
 0015               5 DQ = Y
 0016                 GO TO 10
 0017               7 DQ = 1.D0-Y
 0018                 NCYL=0
 0019              10 DET = DSQRT(-2.D0*DLOG(DQ))
 0020                 DXN = DET-((.010328D0*DET+.802853D0)*DET+2.515517D0)/(((.001308D0
                     1*DET+.189269D0)*DET+1.432788D0)*DET + 1.D0)
 0021                 IF(Y-.5D0)12,12,13
 0022              12 DXN = -DXN
 0023              13 DPN = YORMX(DXN)
 0024                 DER = DPN -Y
 0025                 IF(DABS(DER/Y)-1.D-6)14,14,15
 0026              15 NCYL = NCYL +1
 0027                 IF(NCYL-10)17,17,14
 0028              17 DXP = DXN
 0029              29 DZ = YORMZ(DXP)
 0030                 IF ( DZ)  27,27,28
 0031              27 DXP = DXP/2.D0
 0032                 GO TO 29
 0033              28 DXN = DXN-DER/DZ
 0034                 GO TO 13
 0035              14 YORMP = DXN
 0036              99 RETURN
 0037                 END
```

```
FORTRAN IV G LEVEL  19                YORNZ              DATE = 71056          13/17/47            PAGE 0001

             DOUBLE PRECISION FUNCTION YORNZ(X)
       C     NORMAL ORDINATE
             DOUBLE PRECISION X,Y,YORNZ,EY
             Y=X
             EY = -Y*Y/2.D0
             IF(EY+80.D0)1,1,2
           1 YORNZ = 0.D0
             GO TO 99
           2 YORNZ = .398942280401432678*DEXP(EY)
          99 RETURN
             END
```

```
FORTRAN IV G LEVEL  19                DFP                DATE = 71056        13/17/47             PAGE 0001

0001               DOUBLE PRECISION FUNCTION DFP(DP)
0002               DOUBLE PRECISION DFP,DP
0003               DFP = DLOG(DP/(1.D0-DP))
0004               RETURN
0005               END
```

```
FCRTRAN IV G LEVEL  19            DFZ            DATE = 71056        13/17/47          PAGE 0001

 C001              DOUBLE PRECISICN FUNCTION DFZ(DY)
 CCC2              DOUBLE PRECISICN DFZ,DY,DE
 CCC3              DE = DEXP(DY)
 CC04              DFZ = DE/((1.CC+DE)**2)
 CCC5              RETURN
 CCC6              END
```

```
FORTRAN IV G LEVEL  19                DFX                DATE = 71056          13/17/47              PAGE 0001

 CCC1              DOUBLE PRECISION FUNCTION DFX(DY)
 C0C2              DOUBLE PRECISION DFX,DY,DE
 CC03              DE = DEXP(DY)
 CCC4              DFX = DE/(1.D0+DE)
 CCC5              RETURN
 CC06              END
```

FCRTRAN IV G LEVEL 19 DFP3 DATE = 71056 13/17/47 PAGF 0001

```
      DOUBLE PRECISICN FUNCTION DFP3(DP)
      UOUBLE PRECISICN UFP3,DP
      DFP3 = -DLOG(-CLOG(DP))
      RFTURN
      E.D
```

```
FORTRAN IV G LEVEL  19            DF23            DATE = 71056            13/17/47            PAGE 0001

      DOUBLE PRECISION FUNCTION DF23(DYI)
      DOUBLE PRECISION DF23,DP,DYI
      DP=DEXP(-DEXP(-DYI))
      DF23 = - DP * DLOG(DP)
      RETURN
      END
```

```
FCRTRAN IV G LEVEL  19              DFX3              DATE = 71056        13/17/47              PAGF 0001

                   DOUBLE PRECISICN FUNCTION DFX3(DY)
                   DOUBLE PRECISICN DFX3,DY
                   DFX3 = DEXP(-DEXP(-DY))
                   RETURN
                   END
```

```
FORTRAN IV G LEVEL  19              DFP4              DATE = 71056        13/17/47            PAGE 0001

 0001               DOUBLE PRECISION FUNCTION DFP4(DP)
 0002               DOUBLE PRECISION DFP4,DP
 0003               DFP4 = DARSIN(2.D0 * DP - 1.D0)
 0004               RETURN
 0005               END
```

```
FCRTRAN IV G LEVEL  19              DFZ4              DATE = 71056        13/17/47            PAGE 0001

                DOUBLE PRECISICN FUNCTION DFZ4(DY)
                DOUHLE PRECISICN DFZ4,DY
                DFZ4 = .5D0 * DCOS(DY)
                RETURN
                END
```

```
FCRTRAN IV G LEVEL  19              DFX4            DATE = 71056         13/17/47            PAGE 0001

 CCG1              DOUBLE PRECISICN FUNCTION DFX4(DY)
 C0O2              DOUBLE PRECISICN DFX4,DY
 C0O3              DFX4 = .5D0 * (1.D0 + DSIN(DY))
 CCC4              RETURN
 CCC5              END
```

```
      DOUBLE PRECISION FUNCTION DFP2(DP)
      DOUBLE PRECISION DFP2, DP, DOMEG, DRR, DEX, DLOG
      COMMON NDET(5), X(100), N(100), NR(100), DV1, DV2, DV3,
     1DV4, DV5, DV6, DV7, DVD, DVE, DVA, DVB, NCL, KEY,
     2KK, IFUN, ITRAN, IREC
      DOMEG = 0.9999999999D38
      DFP2 = DOMEG
      IF(DP.GE.1.D0) GO TO 99
      DFP2 = -DOMEG
      IF(DP.LE.0.D0) GO TO 99
      DRR = NR(100)
      DEX = 1.D0/DRR
      DFP2 = (-DRR*DLOG(1.D0-DP))**DEX
   99 RETURN
      END
```

```
      DOUBLE PRECISION FUNCTION DFZ2(DY)
      DOUBLE PRECISION DFZ2, DY, DR, DEXP
      COMMON NDET(5), X(100), N(100), NR(100), DV1, DV2, DV3,
     1DV4, DV5, DV6, DV7, DVD, DVE, DVA, DVB, NCL, KEY,
     2KK, IFUN, ITRAN, IREC
      LR = NR(100)
      DR = LR
      DFZ2 = DY**(LR-1)*DEXP(-DY**LR/DR)
      RETURN
      END
```

```
      DOUBLE PRECISION FUNCTION DFX2(DY)
      DOUBLE PRECISION DFX2, DY, DRR, DEXP
      COMMON NDET(5), X(100), N(100), NR(100), DV1, DV2,
     1DV3, DV4, DV5, DV6, DV7, DVD, DVE, DVA, DVB, NCL,
     2KEY, KK, IFUN, ITRAN, IREC
      DRR = NR(100)
      DFX2 = 1.D0 - DEXP(-DY**NR(100)/DRR)
      RETURN
      END
```

```
0001          SUBROUTINE GRACOF
0002          DIMENSION PCAP(100),PS(100),PCAPS(100),XS(100),IS(100),R(10),P(100
             1)
0003          COMMON NDET(5),X(100),N(100),NR(100),DV1,DV2,DV3,DV4,DV5,DV6,DV7,
             1DVD,DVE,DVA,DVB,NCL,KEY,KK,IFUN,ITRAN,IREC
0004          DATA Q/1H1/,R/2H.1,2H.2,2H.3,2H.4,2H.5,2H.6,2H.7,2H.8,2H.9,2H1./,S
             1/1H-/,STAR/1H*/
0005          CALL DISPLA(DVE)
0006          READ(28'1)P
0007          READ(28'2)PCAP
0008          CALL RESET(DVD)
0009          CALL BLANK
0010          DO 10 I=1,NCL
0011          IS(I) = I
0012          XS(I) = X(I)
0013          PS(I) = P(I)
0014       10 PCAPS(I) = PCAP(I)
0015          NCLM1= NCL - 1
0016          DO 500 I=1,NCLM1
0017          II = I + 1
0018          DO 500 J = II,NCL
0019          IF(XS(I) .LT. XS(J))GO TO 500
0020          STORE = XS(J)
0021          XS(J) = XS(I)
0022          XS(I) = STORE
0023          STORE = PS(J)
0024          PS(J) = PS(I)
0025          PS(I) = STORE
0026          STORE = PCAPS(J)
0027          PCAPS(J) = PCAPS(I)
0028          PCAPS(I) = STORE
0029          ISTORE = IS(J)
0030          IS(J) = IS(I)
0031          IS(I) = ISTORE
0032      500 CONTINUE
0033          XMIN = XS(1)
0034          XMAX = XS(NCL)
0035          XUP = XMIN + 1.1*(XMAX-XMIN)
0036          XLOW = XMIN - 0.1*(XMAX-XMIN)
0037          WRITE(4,1000)
0038          CALL WPFNTS(DVE)
0039     1000 FORMAT('P1'/'PC'/'PO      THE GRAPH BELOW DISPLAYS THE OBSERVED AND
             1 PREDICTED PROPORTION OF '/'P SUCCESS FOR EACH BATCH (CLASS).'/
             2'P       THE HORIZONTAL AXIS IS THE X-DOSAGE OF EACH BATCH (ARRANGE
             2D IN'/'P ASCENDING ORDER).  THE BATCH NUMBER FOR EACH CORRESPONDIN
             3G BATCH IS'/'P DIRECTLY UNDER THE AXIS AND THE TWO LOWER FIGURES A
             4RE THE MINIMUM AND'/ 'P MAXIMUM X-DOSAGE.'/'P       THE VERTICAL AX
             5IS IS THE PROPORTION OF SUCCESSES AXIS.  THE LINED '/'P GRAPH IS T
             7HE PREDICTED PROPORTION OF SUCCESSES AND THE ASTERICKS'/'P DENOTE
             8THE OBSERVED PROPORTION OF SUCCESSES.'/'P       WHEN YOU ARE READY
             9TO PROCEED, DEPRESS PROGRAM FUNCTION KEY 1.')
0040          IBUF = PLOT(DVE)
0041          CALL COORD(XLOW,-.2,XUP,2.)
0042          CALL PLACE(DVC,XLOW,0.)
0043          CALL LINE(DVC,XUP,0.)
0044          CALL PLACE(DVD,XMIN,-.2)
```

```
FORTRAN IV G LEVEL  19              GRACDF              DATE = 71056            13/18/47              PAGE 0002

                  CALL LINE(DVC,XMIN,1.)
                  WRITE(6,110)IBUF
              110 FORMAT(10X,'BUFFER SPACE REMAINING IS',I4)
                  CALL PLACE(DVD,XMIN,0.)
                  DO 200 I=1,NCLM1
                  J = I + 1
                  CALL LINE(DVC,XS(I),PCAPS(I))
                  CALL LINE(DVC,XS(J),PCAPS(I))
              200 CONTINUE
                  CALL LINE(DVC,XS(NCL),PCAPS(NCL))
                  CALL LINE(DVC,XUP,PCAPS(NCL))
                  CALL PLACE(DVD,XS(NCL),PS(NCL))
                  CALL CHAR(DVC,STAR,1,68)
                  DO 201 I=1,NCLM1
                  J = NCL - I
                  CALL PLACE(DVD,XS(J),PS(J))
                  CALL CHAR(DVD,STAR,1,68)
              201 CONTINUE
                  DO 16 J=1,10
                  XJ = FLOAT(J)*.1
                  CALL PLACE(DVC,XLOW,XJ)
                  CALL CHAR(DVC,R(J),2,68)
                  CALL PLACE(DVD,XMIN,XJ)
                  CALL CHAR(DVC,S,1,68)
               16 CONTINUE
                  DO 17 J=1,NCL
                  CALL PLACE(DVC,XS(J),0.)
                  CALL CHAR(DVC,C,1,68)
                  CALL PLACE(DVD,XS(J),-.1)
                  IF(IS(J).LT.10)GO TO 20
                  WRITE(4,40)IS(J)
                  CALL WRFMTS(DVC)
               40 FORMAT('P.',I2)
                  GO TO 17
               20 WRITE(4,41)IS(J)
                  CALL WRFMTS(DVC)
               41 FORMAT('P.',I1)
               17 CONTINUE
                  CALL PLACE(DVD,XS(1),-.2)
                  WRITE(4,42)XS(1)
                  CALL WRFMTS(DVC)
               42 FORMAT('P.',1PE12.4)
                  XSLIM = XS(NCL) - .1*(XMAX - XMIN)
                  CALL PLACE(DVD,XSLIM,-.2)
                  WRITE(4,42)XS(NCL)
                  CALL WRFMTS(DVC)
                  II = SIZE(DVC)
                  IF(II.GT.IBUF)GO TO 50
                  IBUF = PLOT(DVC)
               75 K = DETAIN(NDET)
                  IF(K.NE.1)GO TO 75
                  IF(NDET(4).EQ.31)GO TO 750
                  IF(NDET(4).EQ.30)GO TO 67
                  IF(NDET(4).EQ.1)GO TO 74
                  GO TO 75
              750 KEY = 50
```

```
FORTRAN IV G LEVEL  19            GRACOF            DATE = 71056            13/18/47            PAGE 0003

               GO TO 74
            67 KEY = 1
            50 WRITE(6,501)II
           501 FORMAT(10X,'II = ',I5)
            74 CALL BLANK
               CALL RESET(DVC)
               RETURN
               END
```

```
0001            SUBROUTINE OUTPUA
0002            DIMENSION P(100),PCAP(100),YPR(100),YWORK(100),Y(100),W(100),ER(10
               10),A(10)
0003            DATA A/4H   PR,4HOBIT,4H    L,4HCGIT,4H LOG,4H-LOG,4H ARC,4H SIN,
               14H WEI,4HBULL/
0004            COMMON NDET(5),X(100),N(100),NR(100),DV1,DV2,DV3,DV4,DV5,DV6,DV7,
               1DVD,DVF,DVA,DVP,NCL,KEY,KK,IFUN,ITRAN,IRFC
0005            CALL DISPLA(DVA)
0006            READ(28'1)P
0007            READ(28'2)PCAP
0008            READ(28'3)YPR
0009            READ(28'4)Y
0010            READ(28'5)YWORK
0011            READ(28'6)W
0012            READ(28'7)ER
0013            IXII = 2 * ITRAN
0014            IXI = IXII -1
0015        102 NL= NCL
0016            JKEY = 0
0017            JEY = 0
0018            I = 1
0019            K = 0
0020            NI = 1
0021         94 IF(NL.LT.10)JEY = 1
0022          1 IF(NL.GT.5)GO TO 2
0023            JKEY = 1
0024            NN = NCL
0025            GO TO 3
0026          2 NN = NI + 4
0027          3 KK = FLOAT(K) + .5
0028            KK = KK * 2
0029            IF(KK.EQ.K)GO TO 4
0030          6 WRITE(4,10)(JX,JX=NI,NN)
0031            CALL WRFMTS(DVA)
0032         10 FORMAT('PC'/'P  CLASS',I9,4I12)
0033            WRITE(4,15)(N(JX),JX=NI,NN)
0034            CALL WRFMTS(DVA)
0035         15 FORMAT('P   SIZE',I9,4I12)
0036            WRITE(4,20)(P(JX),JX=NI,NN)
0037            CALL WRFMTS(DVA)
0038         20 FORMAT('P  P(OBS )',1P5E12.4)
0039            WRITE(4,25)(PCAP(JX),JX=NI,NN)
0040            CALL WRFMTS(DVA)
0041         25 FORMAT('P  P(PRED)',1P5E12.4)
0042            WRITE(4,30)(YPR(JX),JX=NI,NN)
0043            CALL WRFMTS(DVA)
0044         30 FORMAT('P  Y(PRED)',1P5E12.4)
0045            WRITE(4,35)(Y(JX),JX=NI,NN)
0046            CALL WRFMTS(DVA)
0047         35 FORMAT('P  Y(OBS )',1P5E12.4)
0048            WRITE(4,40)(YWORK(JX),JX=NI,NN)
0049            CALL WRFMTS(DVA)
0050         40 FORMAT('P  Y(WORK)',1P5E12.4)
0051            WRITE(4,45)(W(JX),JX=NI,NN)
0052            CALL WRFMTS(DVA)
0053         45 FORMAT('P  WEIGHTS',1P5E12.4)
```

```
      WRITE(4,50)(X(JX),JX=NI,NN)
      CALL WREMTS(DVA)
   50 FORMAT('PO XDOSAGE',1P5E12.4)
      WRITE(4,55)(ER(JX),JX=NI,NN)
      CALL WREMTS(DVA)
   55 FORMAT('P  Y-ERROR',1P5E12.4)
      IF(KK.NE.K .OR. JKEY.EQ.1)GO TO 9
   11 NI=NI + 5
      K = K + 1
      NL= NL- 5
      IF(JKEY)94,94,74
    9 IBUF = PLOT(DVA)
  100 J = OBTAIN(NDET)
      CALL RESET(DVA)
      IF(J.NE.1)GO TO 100
      IF(NDET(4).EQ.31)GO TO 750
      IF(NDET(4).EQ.30)GO TO 67
      IF(NDET(4).EQ.1)GO TO 101
      IF(NDET(4).EQ.4)GO TO 102
      GO TO 100
  101 CALL BLANK
      GO TO 11
    4 I2 = I + 1
      CALL BLANK
      WRITE(4,5)A(IXI),A(IXII),I
      CALL WREMTS(DVA)
    5 FORMAT('P1'/'PC',20X,2A4,' ANALYSIS',23X,'PAGE ',I3)
      IF( JEY.EQ.1)GO TO 12
      WRITE(4,60)I2
      CALL WREMTS(DVA)
   60 FORMAT('P       TO SEE PAGE ',I3,' DEPRESS PROGRAM FUNCTION KEY 1')
      I = I + 1
      GO TO 6
   12 WRITE(4,65)
      CALL WREMTS(DVA)
   65 FORMAT('P       THIS IS THE LAST PAGE OF THIS SECTION OF YOUR OUTPU
     1T.'/'P IF YOU WISH TO REVIEW THIS SECTION, DEPRESS PROGRAM FUNCTIO
     2N KEY 4'/'P OTHERWISE, DEPRESS PROGRAM FUNCTION KEY 1.')
      GO TO 6
  750 KEY = 50
      GO TO 74
   67 KEY = 1
   74 CALL BLANK
      CALL RESET(DVA)
      RETURN
      END
```

```
FORTRAN IV G LEVEL  1S                 OUTPUT             DATE = 71188          10/48/21

0001            SUBROUTINE OUTPUT
0002            DIMENSION A(10),Y(100)
0003            COMMON NDET(5),X(100),N(100),NR(100),DV1,DV2,DV3,DV4,DV5,DV6,DV7,
               1DVC,DVF,DVA,DVB,NCL,KEY,KK,IFUN,ITRAN,IREC
0004            DATA STAR/1H*/,O/1H)/,S/1H-/,A/4H  PR,4HOBIT,4H   L,4HOGIT,4H LOG,
               14H-LOG,4H ARC,4H SIN,4H WEI,4HBULL/
0005            CALL DISPLA(DVB)
0006            IF(KEY.NE.25)GO TO 1000
0007            READ(28'10)X
0008            READ(28'11)N
0009            READ(28'12)NR
0010            READ(28'13)NCL,ITRAN
0011            KEY = 0
0012       1000 READ(28'4)Y
0013       2000 READ(28'9)ALF,BETA
0014       2001 READ(28'8)SSR,SSE
0015            IXI = 2 * ITRAN - 1
0016            IXII = IXI + 1
0017            JR = 1
0018            JE = NCL -2
0019            JTOT = JR + JE
0020            IF(KEY.EQ.6)GO TO 491
0021            VSE = SSE / FLOAT(JE)
0022            STOT = SSR + SSE
0023            F = SSR / VSE
0024            WRITE(4,71)A(IXI),A(IXII),SSR,JR,SSR,SSE,JE,VSE,STOT,JTOT,F
0025            CALL WRFMTS(DVB)
0026         71 FORMAT('P1'/'P0',27X,2A4,' ANALYSIS'/'P0',23X,'ANALYSIS OF VARIANC
               5E TABLE'/'P0',21X,'SUM OF',11X,'DEGREES OF',12X,'MEAN'/'P   SOURCE
               6',12X,'SQUARES',12X,'FREEDOM',13X,'SQUARE'/'P0REGRESSION',5X,E18.8
               7,10X,I5,5X,E18.8/'P0ERROR',12X,E16.8,10X,I5,7X,E16.8/'P0TOTAL',10X
               8,E18.8,10X,I5/'P0',21X,'CALCULATED F = ',E16.8)
0027            WRITE(4,72)JE
0028            CALL WRFMTS(DVB)
0029         72 FORMAT('P0'/'P0      IF YOU WISH TO LEAVE THE QUANTAL PROGRAM TO DE
               1TERMINE F(1,',I2,'),8Y'/'P CALLING THE "CALCG" PROGRAM, DEPRESS PR
               2OGRAM FUNCTION KEY 11.  IF YOU'/'P WISH TO CONTINUE, DEPRESS PROGR
               3AM FUNCTION KEY 1.')
0030            TRUE = PLOT(DVB)
0031         73 I = DETAIN(NDET)
0032            IF(I.NE.1)GO TO 73
0033            IF(NDET(4).EQ.31)GO TO 750
0034            IF(NDET(4).EQ.30)GO TO 67
0035            IF(NDET(4).EQ.11)GO TO 8C
0036            IF(NDET(4).NE.1)GO TO 73
0037        491 CALL BLANK
0038            CALL RESET(DVB)
0039            WRITE(4,400)A(IXI),A(IXII),ALF,BETA
0040            CALL WRFMTS(DVB)
0041        400 FORMAT('P1'/'P0      THE FOLLOWING IS THE GRAPH OF THE STRAIGHT LIN
               1E Y = ALPHA + BETA * X'/'P WHERE ALPHA AND BETA HAVE BEEN ESTIMATE
               2D IN THE ',2A4, ' ANALYSIS.'/'P HERE THE EQUATION IS Y = ',F16.6,
               3' + ',E16.6,' * X.'/'P THE HORIZONTAL AXIS IS THE DOSAGE-AXIS AND
               4THE VERTICAL AXIS IS THE Y-AXIS'/'P THE VALUES DENOTED BY '*'' AR
               5E THE OBSERVED Y-VALUES')
0042       2002 WRITE(4,2004)
```

```
0043            CALL WRFMTS(DVB)
0044      *  2004 FORMAT('PO     WHEN YOU ARE READY TO PROCEED, DEPRESS PROGRAM FUNC
               1TION KEY 1.')
0045            XMAX = X(1)
0046            YMAX = Y(1)
0047            DO 4 I=2,NCL
0048          * IF(X(I)-XMAX)5,5,6
0049          6 XMAX = X(I)
0050          5 IF(Y(I) - YMAX)4,4,7
0051          7 YMAX = Y(I)
0052          4 CONTINUE
0053            XMIN = X(1)
0054            YMIN = Y(1)
0055            DO 8 I=2,NCL
0056            IF(X(I)-XMIN)9,10,10
0057         9  XMIN = X(I)
0058         10 IF(Y(I) - YMIN)11,8,8
0059         11 YMIN = Y(I)
0060          8 CONTINUE
0061            XLGTH = XMAX - XMIN
0062            XINC = -.08 * XLGTH
0063            YLGTH = YMAX - YMIN
0064            YINC = -.05 * YLGTH
0065            IF(XMIN.GT.0.)XMIN = XINC
0066            IF(XMAX.LT.0.)XMAX=0.
0067            IF(YMIN.GT.0.)YMIN = YINC
0068            IF(YMAX.LT.0.)YMAX=0.
0069         85 Y2MIN = 1.2 * YMIN
0070            Y2MAX = 2.8 * YMAX
0071            X2MAX = XMAX - XINC
0072            CALL UCORD(XMIN,Y2MIN,X2MAX,Y2MAX)
0073            CALL PLACE(DVB,0.,YMIN)
0074            CALL LINE(DVB,0.,YMAX)
0075            CALL PLACE(DVB,XMIN,0.)
0076            CALL LINE(DVB,XMAX,0.)
0077         13 IBUF = PLOT(DVB)
0078            WRITE(6,200)IBUF
0079        200 FORMAT(10X,'BUFFER SPACE REMAINING IS',I4)
0080         14 XLGTH = ((X2MAX-XMIN)/200.)
0081            IF(KEY.EQ.6)GO TO 270
0082            DO 15 IP=1,201
0083            XPM1 = FLOAT(IP-1)
0084            XPLOT = XMIN + XPM1 * XLGTH
0085            YPLOT = ALF + BETA * XPLOT
0086            IF(YPLOT.LT.Y2MIN .OR. YPLOT.GT.Y2MAX)GO TO 15
0087            CALL POINT(DVB,XPLOT,YPLOT)
0088         15 CONTINUE
0089        270 DO 16 I=1,NCL
0090            CALL PLACE(DVB,X(I),Y(I))
0091            CALL CHAR(DVB,STAR,1,69)
0092         16 CONTINUE
0093            XX = XMIN
0094            XNUM = (XMAX - XMIN) * .1
0095            JEY = 1
0096            GO TO 26
0097         36 XX = YMIN
```

```
      XNUM = (YMAX - YMIN) * .1
   26 DO 24 I=1,10
      XX = XX + XNUM
      GO TO (27,28),JEY
   27 CALL PLACE(DVB,XX,0.)
      CALL CHAR(DVB,0,1,69)
      CALL PLACE(DVB,XX,YINC)
      GO TO 29
   28 CALL PLACE(DVB,0.,XX)
      CALL CHAR(DVB,S,1,69)
      CALL PLACE(DVB,XINC,XX)
   29 IF(XX.LE.-1000.)GO TO 30
      IF(XX.LE.-100.)GO TO 31
      IF(XX.LE.-10.)GO TO 32
      IF(XX.LT.0.)GO TO 33
      IF(XX.LT.10.)GO TO 34
      IF(XX.LT.100.)GO TO 33
      IF(XX.LT.1000.)GO TO 32
      IF(XX.LT.10000.)GO TO 31
   30 WRITE(4,35)XX
      CALL WRFMTS(DVB)
   35 FORMAT('P.',F9.2)
      GO TO 24
   31 WRITE(4,40)XX
      CALL WRFMTS(DVB)
   40 FORMAT('P.',F7.2)
      GO TO 24
   32 WRITE(4,45)XX
      CALL WRFMTS(DVB)
   45 FORMAT('P.',F6.2)
      GO TO 24
   33 WRITE(4,55)XX
      CALL WRFMTS(DVB)
   55 FORMAT('P.',F5.2)
      GO TO 24
   34 WRITE(4,60)XX
      CALL WRFMTS(DVB)
   60 FORMAT('P.',F4.2)
   24 CONTINUE
      IF(JEY.NE.1)GO TO 22
      JEY = 2
      GO TO 36
   22 II = SIZE(DVB)
      IF(II.GT.IBUF)GO TO 50
      IBUF = PLOT(DVB)
   50 WRITE(6,300)II
  300 FORMAT(10X,'II = ',I5)
   75 K = DETAIN(NDET)
      IF(K.NE.1)GO TO 75
      IF(NDET(4).EQ.31)GO TO 750
      IF(NDET(4).EQ.30)GO TO 67
      IF(NDET(4).EQ.1)GO TO 99
      GO TO 75
   80 WRITE(28'10)X
      WRITE(28'11)N
      WRITE(28'12)AR
```

FORTRAN IV G LEVEL 19 OUTPUT DATE = 71188 10/48/21

```
      WRITE(28'13)ACL,ITRAN
      KEY = 50
      GO TO 99
  750 KEY = 50
      GO TO 99
   67 KEY = 1
   99 CALL BLANK
      CALL RESET(DVB)
      RETURN
      END
```

```
FORTRAN IV G LEVEL  19                CHNGFN              DATE = 71056          13/18/23                PAGE 0001

 0001               SUBROUTINE CHNGFN
 0002               COMMON NDET(5),X(100),N(100),NR(100),DV1,DV2,DV3,DV4,DV5,DV6,DV7,
                   1DVD,DVF,DVA,DVB,NCL,KEY,KK,IFUN,ITRAN,IREC
 0003               CALL DISPLA(DVA,DVB,DVE)
 0004               CALL RESET(DVB)
 0005               CALL BLANK
 0006               WRITE(4,5)
 0007               CALL WREMTS(DVB)
 0008             5 FORMAT('PI'/'PC      IF YOU WISH TO ALTER YOUR X-DOSAGE BY ONE OF T
                   1HE FOLLOWING FUNCTIONS' /'P DEPRESS PROGRAM FUNCTION KEY 3.  OTHER
                   2WISE, DEPRESS PROGRAM FUNCTION'/'P KEY 1.'/'PO'/'PO1.  SIN(X)  -
                   3SINE',16X,'7.  EXP(X)  -  EXPONENTIAL'/'P 2.  COS(X)  -  COSINE',1
                   44X,'8.  ALOG(X)  -  NATURAL LOGARITHM'/'P 3.  TAN(X)  -  TANGENT',
                   515X,'9.  ALOG10(X)  -  COMMON LOGARITHM'/'P 4.  ARSIN(X)  -  ARC S
                   6INE',10X,'10. SQRT(X)  -  SQUARE ROOT'/'P 5.  ARCOS(X)  -  ARC COS
                   7INE', 8X,'11. ABS(X)  -  ABSOLUTE VALUE'/'P 6.  ATAN(X)  -  ARC TA
                   8NGENT', 8X,'12. X**2  -  X SQUARED')
 0009               IBUF = PLOT(DVB)
 0010             1 J = DETAIN(NDET)
 0011               IF(J.NE.1)GO TO 1
 0012               IF(NDET(4).EQ.31)GO TO 750
 0013               IF(NDET(4).EQ.30)GO TO 99
 0014               IF(NDET(4).EQ.1)GO TO 50
 0015               IF(NDET(4).EQ.3)GO TO 10
 0016               GO TO 1
 0017            10 WRITE(4,2)
 0018               CALL WREMTS(DVA)
 0019             2 FORMAT('PC      TYPE IN THE NUMBER CORRESPONDING TO THE FUNCTION YO
                   1U DESIRE.    '/'L.',3X)
 0020               IBUF = PLOT(DVA)
 0021               CALL CURSS(DVA,68)
 0022             3 J = DETAIN(NDET)
 0023               IF(J.EQ.1 .AND. NDET(4).EQ.31)GO TO 750
 0024               IF(J.EQ.1 .AND. NDET(4).EQ.30)GO TO 99
 0025               IF(J.NE.2)GO TO 3
 0026               CALL SCIDM(DVA)
 0027               READ(4,15)XFUN
 0028            15 FORMAT(/G3.0)
 0029               CALL BUFRS
 0030               IFUN = XFUN
 0031               IF(IFUN.LT.0 .OR. IFUN.GT.12)GO TO 20
 0032               GO TO(53,53,53,53,53,53,53,1050,1050,1050,53,53),IFUN
 0033            53 KEY = 6
 0034               GO TO 100
 0035            20 K = UNPLOT(DVA)
 0036               CALL RESET(DVA)
 0037               WRITE(4,6)
 0038               CALL WREMTS(DVE)
 0039             6 FORMAT('POTHE NUMBER MUST BE BETWEEN 1 AND 12')
 0040               IBUF = PLOT(DVE)
 0041               CALL RESET(DVE)
 0042               GO TO 10
 0043          1050 DO 55 I=1,NCL
 0044               IF(X(I).LT.0.0)GO TO 52
 0045            55 CONTINUE
 0046               GO TO 53
```

```
0047        52 CALL RESET(DVA)
0048           K = UNPLCT(DVA)
0049           WRITE(4,51)
0050           CALL WRFMTS(CVA)
0051        51 FORMAT('PO THIS FUNCTION CANNOT BE EMPLOYED BECAUSE ONE OF THE X-D
              1IMAGES IS NEGATIVE'/'P IF YOU WISH TO CHOOSE ONE OF THE OTHER FUNC
              2TIONS, DEPRESS PROGRAM'/'P FUNCTION KEY 3.  OTHERWISE DEPRESS PROG
              3RAM FUNCTION KEY 1.')
0052           IBUF = PLCT(DVA)
0053           CALL RESET(DVA)
0054           GO TO 1
0055       750 KEY = 50
0056           GO TO 100
0057        99 KEY = 1
0058           GO TO 100
0059        50 KEY = 5
0060       100 CALL BLANK
0061           CALL RESET(DVA,DVB,DVC)
0062           RETURN
0063           END
```

```
FORTRAN IV G LEVEL  19                 FNS                DATE = 71056          13/18/39              PAGE 0001

0001                SUBROUTINE FNS
0002                COMMON NDET(5),X(100),N(100),NR(100),DV1,DV2,DV3,DV4,DV5,DV6,DV7,
                   1DVD,DVE,DVA,DVB,NCL,KEY,KK,IFUN,ITRAN,IREC
0003                DO 4 I=1,NCL
0004                GO TO(21,22,23,24,25,26,27,28,29,30,31,32),IFUN
0005             21 X(I) = SIN(X(I))
0006                GO TO 4
0007             22 X(I) = COS(X(I))
0008                GO TO 4
0009             23 X(I) = TAN(X(I))
0010                GO TO 4
0011             24 X(I) = ARSIN(X(I))
0012                GO TO 4
0013             25 X(I) = ARCOS(X(I))
0014                GO TO 4
0015             26 X(I) = ATAN(X(I))
0016                GO TO 4
0017             27 X(I) = EXP(X(I))
0018                GO TO 4
0019             28 X(I) = ALOG(X(I))
0020                GO TO 4
0021             29 X(I) = ALOG10(X(I))
0022                GO TO 4
0023             30 X(I) = SQRT(X(I))
0024                GO TO 4
0025             31 X(I) = ABS(X(I))
0026                GO TO 4
0027             32 X(I) = (X(I))**2
0028              4 CONTINUE
0029                RETURN
0030                END
```